

Automatic Lecture Recording

Inauguraldissertation
zur Erlangung des akademischen Grades
eines Doktors der Naturwissenschaften
der Universität Mannheim

Vorgelegt von
M. Sc. Diplom-Inform. (FH)
Fleming Helge Lampi
aus
Heidelberg

Karlsruhe, 2010

Dekan: Professor Dr. Felix Freiling, Universität Mannheim
Referent: Professor Dr. Wolfgang Effelsberg, Universität Mannheim
Korreferent: Professor Dr. Ralf Steinmetz, Technische Universität Darmstadt

Tag der mündlichen Prüfung: 16. Juli 2010

Acknowledgments

I would like to thank a number of people who have supported me during my years of study and promotion.

At first, I would like to thank my professors at the University of Applied Sciences in Karlsruhe for their work during my diploma and master studies, emphasizing Prof. Dr. Peter A. Henning for having been the advisor of my two theses and for having me introduced to Professor Effelsberg. I would like to thank Professor Dr. Wolfgang Effelsberg for being my doctoral advisor, and inspiring me by many good and fruitful conversations with him and my colleagues. Thanks to all of them for their collaboration.

Out of my colleagues, I especially like to thank Dr. Nicolai Scheele, Dr. Stephan Kopf, Dr. Thomas King, Hendrik Lemelson, and Dr. Tanja Mangold for their support and all the publications written jointly. In addition, I want to thank Manuel Benz and Patric Herweh for their really good work and Eva Haas for proofreading my dissertation.

Furthermore, I would like to thank Jochen Braitinger for supporting me in getting detailed information about the work-flows in broadcast stations, perfecting my education concerning media production.

I would like to thank my family, in particular my aunt Alma Lampi and my mother Helga Lampi, for supporting me during all the years. Finally, I like to thank my girl friend Marion Wehde for her support, her patience and her love during all our years and my studies. Even more, I thank her very much for proofreading my diploma thesis, my master thesis and my dissertation and her reliable strong support even in difficult situations.

Abstract

Lecture recording has become a very common tool to provide students with additional media for their examination preparations. While its effort has to stay reasonable, only a very basic way of recording is done in many cases. Therefore, watching the resulting videos can get very boring completely independent of how interesting the original topic or session was.

This thesis proposes a new approach to lecture recordings by letting distributed computers emulate the work of a human camera team, which is the natural way of creating attractive recordings.

This thesis is structured in six chapters, starting with the examination of the current situation, and taking its constraints into account. The first chapter concludes with a reflection on related work.

Chapter two is about the design of our prototype system. It is deduced from a human camera team in the real world which gets transferred into the virtual world. Finally, a detailed overview about all parts necessary for our prototype and their planned functionality is given. In chapter three, the implementation of all parts and tasks and the incidents occurring during implementation are described in detail.

Chapter four describes the technical experiences made with the different parts during development, testing and evaluation with a view to functionality, performance, and an proposal towards future work. The evaluation of the whole system with students is presented and discussed in the fifth chapter.

Chapter six concludes this thesis by summing up the facts and gives an outlook on future work.

Zusammenfassung

Vorlesungsaufzeichnungen sind mittlerweile ein häufig verwendetes Mittel, um Studierende mit zusätzlichem Material für ihre Prüfungsvorbereitungen auszustatten. Dabei muss der benötigte Aufwand im Verhältnis bleiben, so dass oft nur eine sehr grundlegende und einfache Art der Aufzeichnung realisiert wird. Die daraus entstehenden Resultate zu betrachten kann sehr langweilig werden, unabhängig davon wie interessant das ursprüngliche Thema oder die Vorlesung war.

Die vorliegende Dissertation schlägt einen neuen Ansatz für Vorlesungsaufzeichnungen vor, in dem ein verteiltes Computersystem die Arbeit und Vorgehensweise eines menschlichen Kamerateams nachahmt, um auf diese Weise attraktive Aufzeichnungen herzustellen.

Diese Arbeit ist in sechs Kapitel gegliedert und beginnt mit der Betrachtung der aktuellen Situation und der gegebenen Vorgaben. Das erste Kapitel schliesst mit einer Betrachtung von verwandten Arbeiten.

Kapitel zwei beschreibt das Design des Prototyp-Systems, das von einem menschlichen Kamerateam in der realen Welt abgeleitet und in die virtuelle Welt transferiert wird. Es endet mit einer detaillierten Übersicht über die für den Prototyp notwendigen Teile und ihrer geplanten Funktionalität. Kapitel drei beschreibt im Detail die Implementierung aller Teile und Ihrer Aufgaben, sowie die Besonderheiten, die während der Implementierung aufgetreten sind.

Kapitel vier beinhaltet die technischen Erfahrungen die mit den einzelnen Teilen des Systems während ihrer Entwicklung, ihrer Testphasen und der Beurteilung ihrer Fähigkeiten gemacht wurden, insbesondere mit einem Fokus auf deren Funktionalität und Performanz sowie einem Vorschlag für zukünftige Implementierungen. Die Evaluierung des gesamten Systems mit Studierenden wird in Kapitel fünf detailliert beschrieben und diskutiert.

Kapitel sechs beschliesst diese Arbeit indem es die Fakten zusammenfasst und einen Ausblick auf zukünftige Arbeiten gibt.

Content

1.	Introduction.....	1
1.1.	Basic Idea.....	1
1.2.	Starting Point	2
1.2.1.	Traditional Lectures Today.....	3
1.2.2.	Room for Improvement.....	4
1.3.	Constraints	5
1.3.1.	Types of Recording.....	6
1.3.2.	Aesthetic Considerations	8
1.3.3.	Financial Constraints	9
1.4.	Related Work	10
1.4.1.	Surveillance Recording.....	11
1.4.2.	Meeting Recording.....	12
1.4.3.	Documentary Recording	13
1.4.4.	Presentation Recording	14
1.4.5.	Lecture Recording.....	17
1.4.6.	Additional Related Work	21
2.	Design of the Distributed System	25
2.1.	Analyzing the Real World	25
2.1.1.	Determining the “Ingredients”	25
2.1.2.	Details of a Real Camera Team	28
2.1.3.	Important Constraints.....	35
2.2.	Determining the Parts of our System	37
2.2.1.	The Director	38
2.2.2.	The Cameraman	39
2.2.3.	Sensor Tools.....	41
2.2.4.	The Sound Engineer.....	46
2.2.5.	The Lighting Technician	48
2.2.6.	The Audio/Video Mixing Console.....	48
2.3.	System Overview	49
2.3.1.	Virtual Director’s Main Ideas	51
2.3.2.	Virtual Cameraman’s Main Ideas	53
2.3.3.	Sensor Tools’ Main Ideas	53

2.3.4.	Virtual Sound Engineer's Main Ideas	54
2.3.5.	AV Mixing Console's Main Ideas	55
3.	System Implementation	57
3.1.	Director Module.....	57
3.1.1.	Tasks to Fulfill	60
3.1.2.	Implementation Details	63
3.2.	Cameraman Module.....	74
3.2.1.	Tasks to Fulfill	74
3.2.2.	Implementation Details	76
3.3.	Sensor Tools Module	87
3.3.1.	WLAN Indoor Positioning System	88
3.3.2.	Question Management Software	91
3.4.	Sound Engineer	98
3.4.1.	Tasks to fulfill	98
3.4.2.	Implementation Details	101
3.5.	Audio-Video Mixer/Recorder	105
3.5.1.	Tasks to Fulfill	107
3.5.2.	Implementation Details	110
4.	Technical Experience.....	117
4.1.	Experience with the director module	117
4.1.1.	Evaluation of the Virtual Director	117
4.1.2.	Testing Setup	119
4.1.3.	Simulation Results	121
4.1.4.	Overall Performance	122
4.2.	Experience with the cameraman module	126
4.2.1.	Performance of the image processing algorithms	126
4.2.2.	Performance of the camera controlling algorithms.....	129
4.2.3.	Overall Performance	132
4.3.	Experience with the Sensor Tools module.....	133
4.3.1.	Experience with the Question – Answer Interaction Control	133
4.3.2.	Experiences with the Event Reporting.....	135
4.3.3.	Experiences with the Audio Streaming.....	136
4.3.4.	Overall Performance	138

4.4.	Experiences with the AV Mixer/Recorder.....	138
4.4.1.	Experience with the AV Decoding	138
4.4.2.	Experiences with the Output Trigger.....	139
4.4.3.	Experiences with the Video Processing.....	140
4.4.4.	Experience with the AV Output.....	141
4.4.5.	Overall Performance	142
4.5.	Experience with the Sound Engineer module	142
4.5.1.	Audio Mixing and Mastering.....	142
4.5.2.	Overall Performance	146
4.5.3.	Improving the AV Mixer/Recorder	146
5.	Evaluation with Students	149
5.1.	Evaluation Description.....	149
5.1.1.	Evaluation Design.....	149
5.1.2.	Description of the sample and its participants	154
5.1.3.	Operationalization of the constructs	154
5.1.4.	Presentation of the evaluation method.....	159
5.2.	Evaluation Results	159
5.3.	Discussion of the evaluation	165
6.	Summary	167
6.1.	The Virtual Camera Team	167
6.2.	Implementation Experiences.....	168
6.3.	Evaluation Experience	169
6.4.	Rating the Prototype	170
6.4.1.	Aesthetic Approach.....	170
6.4.2.	Affordable Approach	171
6.4.3.	Space- and Time-Saving Approach	171
6.4.4.	Successful Prototype	172
6.5.	Outlook	173
6.5.1.	Improving the Current Prototype	173
6.5.2.	Extending the Current Prototype	173
6.5.3.	Transferring Automatic Lecture Recording to other environments...	173
7.	Appendix.....	173
7.1.	Configuration Files	173

7.1.1.	XML file of the FSM used in our prototype	173
7.1.2.	Example Configuration File of the Cameraman	173
7.2.	Sourcecode Snippets	173
7.2.1.	Function “FSM.GetTimerInterval”	173
7.2.2.	Procedure “AVMixer.startgettingFrames”	173
7.2.3.	Function “DefineCompressionLine”	173
7.2.4.	Function “Sample2DB”	173
7.2.5.	Function “DB2SampleValue”	173
7.2.6.	Function “WaveExtrema”	173
7.2.7.	Function “getFactor”	173
7.3.	Original Evaluation Papers	173
7.3.1.	German Pre-Test Form	173
7.3.2.	German Questionnaire	173
7.3.3.	German Post-Test Form	173
8.	Bibliography	173

List of Figures

Figure 1: Invoice snippet of a live production.....	10
Figure 2: Part of a Storyboard of "Kaffee oder Tee" of the SWR	29
Figure 3: Example of wrong recording positions according to the "Line of Action" ..	30
Figure 4: Example of correct recording positions according to the "Line of Action" .	31
Figure 5: Schematic view of equipment for studio production (based on Schmidt, 2005)	33
Figure 6: The job of a cameraman as a work-flow	41
Figure 7: Interaction Diagram of Lecturer and Questioner	44
Figure 8: Systems and Communications Channels Overview	49
Figure 9: Draft of the Prototype System	50
Figure 10: Graph of an Exemplary FSM for Standard Lectures.....	52
Figure 11: Three exemplary exposures (Benz, 2007).....	77
Figure 12: Skin color detection example (Benz, 2007).	78
Figure 13: Images of the steps in background subtraction (Benz, 2007).....	78
Figure 14: Skin detection and region joining example (Benz, 2007).	79
Figure 15: Results of two backlight compensation algorithms (Benz, 2007).....	79
Figure 16: Comparing annotation visibility of color and gray scale images.	80
Figure 17: Distance of two points in the RGB color space.....	81
Figure 18: Frame arrangement tests of abstracted protagonist (Benz, 2007).	83
Figure 19: Lecturer and Questioner aligned in a shot - counter-shot scenario.	83
Figure 20: Motion-triggered automatic zoom-out followed by automatic zoom-in (Benz, 2007).....	84
Figure 21: Information exchange from director to camera via the cameraman and back (based on Benz, 2007).....	84
Figure 22: Estimated area marked and seat of student confirmed.	90
Figure 23: Standard interface of the questioners' client.....	92
Figure 24: Popup on the Lecturer's computer showing announced questions.	94
Figure 25: Basic question - answer interaction, amended with client screen shots.	95
Figure 26: GUI of the QM server, client on seat 77 asking.	98
Figure 27: Ideal silence (top), minimal noise as silence (middle), and this minimal noise after normalization (bottom).....	100
Figure 28: Part of a 440Hz sine tone at 48 kHz (top) and at 8 kHz (bottom).....	104

Figure 29: Screen shot of the status display of the AV Mixer/Recorder.	109
Figure 30: Overview of the AV Mixer/Recorder.	110
Figure 31: Percentage of fulfilled requested shots after n seconds.	122
Figure 32: Explanation of an EDL entry.	124
Figure 33: Exemplary status message displays of three virtual cameramen.	132
Figure 34: Screen shot of an exemplary curve of a combined noise-gate, compressor, and limiter. Input-dB on the x-axis, output-dB on the y-axis.	144
Figure 35: A 440 Hz sine curve with different amplification factors in adjacent audio frames.	145
Figure 36: Translated knowledge pre-test.	151
Figure 37: Pages 1 and 2 of the translated questionnaire.	153
Figure 38: Pages 3 and 4 of the translated questionnaire.	153
Figure 39: Average values of attentiveness and interest in the video compared.	161
Figure 40: Average values of three QCM aspects compared.	162
Figure 41: Average values of learning gain compared.	163

List of Tables

Table 1: Calculation spreadsheet	36
Table 2: Aggregated Actions Controlling the Interaction.....	45
Table 3: Resulting weights of the positions inside the states' history buffer.	71
Table 4: Mapping of camera status messages to factors.....	72
Table 5: Iris control: f values, ratio of light and camera parameters; part 1.....	76
Table 6: Iris control: f values, ratio of light and camera parameters; part 2.....	76
Table 7: Exemplary stage directions from the director module to the cameraman module.....	85
Table 8: Parameters and possible values of the camera interface.....	86
Table 9: Exemplary sensor inputs with timestamps of the “average lecture”.	119
Table 10: Simulation results of both finite state machines.	121
Table 11: Differences between the tested two types of lecture recording.	152
Table 12: Items of the construct "attentiveness" (T. Mangold).	156
Table 13: Items of the construct "interest in the video" (T. Mangold).	157
Table 14: Items of the construct "motivation" (T. Mangold based on Rheinberg, Vollmeyer & Burns, 2001).	158
Table 15: t-test results for the construct "attentiveness".....	160
Table 16: t-test results for the construct "interest in the video".....	160
Table 17: t-test results for the construct "QCM interest".....	161
Table 18: t-test results for the construct "QCM probability of success".	161
Table 19: t-test results for the construct "QCM anxiety".....	162
Table 20: t-test results for the construct "learning gain".	163
Table 21: Statistic results of the items concerning the motivation.....	164
Table 22: Assumed number of repetitions of lecture recordings.....	165
Table 23: Assumed learning gain.	165

List of Definitions and Formulas

Definition/Formula 1: Definition of a Finite State Machine.	60
Definition/Formula 2: Definition of an Extended Finite State Machine.	61
Definition/Formula 3: Definition of the Transition - Tuple.....	63
Definition/Formula 4: Formula to calculate weights of the history buffer.	70
Definition/Formula 5: CameraParameter based on f-number of the iris.	76
Definition/Formula 6: Length of a 3D vector; here the distance between the two points named p and q.	81
Definition/Formula 7: Calculation of the audio break duration in case of a UDP packet loss.	102
Definition/Formula 8: Calculating the amplifying factor including 3 dB headroom.	103
Definition/Formula 9: Translation and Scaling by homogeneous matrices.....	115
Definition/Formula 10: ColorMatrix for changing the transparency of an image.....	115
Definition/Formula 11: Running Gaussian Average formula to update the background model.....	128
Definition/Formula 12: Normalizing red and green values for skin color detection.	128
Definition/Formula 13: Calculating the zoom factor to frame a questioner.	130
Definition/Formula 14: Calculating the zoom factor for a distance of four m.	130

List of Abbreviations

°	– Degree
3D	– Three dimensions / Three dimensional
802.11	– Number of the IEEE norm of wireless local area networks, synonym for WLAN
ALR	– Automatic Lecture Recording
AP	– Access Point
API	– Application Programming Interface
AV	– Audio / Video
AVI	– Audio Video Interleaved
BBC	– British Broadcasting Corporation
BMP	– File-extension of a saved bitmap object
BR	– Bayerischer Rundfunk
CF	– Compact Flash
CODEC	– Coder / Decoder
CPU	– Central Processing Unit
dB	– Decibel
DI	– Direct Input
DLL	– Dynamic Link Library
DoS	– Denial of Service
DV	– Digital Video
EDL	– Edit Decision List
EFSM	– Extended Finite State Machine
FIFO	– First In – First Out
fps	– frames per second
FSM	– Finite State Machine
GB	– Giga byte = $1024 * 1024 * 1024$ bytes
GHz	– Giga Hertz = $1000 * 1000 * 1000$ Hertz
GOP	– Group of Pictures
GPS	– Global Positioning System
GUI	– Graphical User Interface
HCI	– Human-Computer Interface

HD	– High Definition (either 1080x720 pixel or 1920x1080 pixel)
HiFi	– High Fidelity
HR	– Hessischer Rundfunk
HTTP	– HyperText Transfer Protocol
Hz	– Hertz – the unit to measure frequencies
ICMP	– Internet Control Message Protocol
IP	– Internet Protocol
JFIF	– JPEG File Interchange Format
JPEG	– Joint Photographic Experts Group
kHz	– kilo Hertz = 1,000 Hertz
LAN	– Local Area Network
LMS	– Learning Management System
MB	– Mega byte = 1024 * 1024 byte
MJPEG	– Motion JPEG
MPEG	– Motion Picture Experts Group
ms	– Millisecond – 1/1000 of a second
MSDN	– Microsoft Developer Network
MTU	– Maximum Transmission Unit
NTSC	– National Television System Committee (also used as abbreviation of SD TV norm, used e.g., in the United States of America)
PAL	– Phase Alternating Line (SD TV norm, used e.g., in Germany)
PCM	– Pulse Code Modulation
PDA	– Personal Digital Assistant
PTZ	– Pan - Tilt - Zoom
QA	– Question-Answer
QBIC	– Q-Belt Integrated Computer
QCM	– Questionnaire to assess Current Motivation
QM	– Question Management
PiP	– Picture-in-Picture
RAID	– Redundant Array of Independent Disks
RAM	– Random Access Memory
RGB	– Red Green Blue
RTCP	– Real Time Control Protocol

RTP	– Real Time Transport Protocol
RTSP	– Real Time Streaming Protocol
SD	– Standard Definition (720x576 pixel for PAL/SECAM, 720x480 pixel for NTSC)
SDI	– Serial Digital Interface
SECAM	– Séquentiel couleur à mémoire (SD TV norm, used e.g., in France)
SMPTE	– Society of Motion Picture and Television Engineers
SWR	– Südwest Rundfunk
SRT	– Schule für Rundfunktechnik, Nuremberg, Germany; meanwhile ARD.ZDF medienakademie
TCP	– Transmission Control Protocol
TV	– Television
UDP	– User Datagram Protocol
UK	– United Kingdom
URL	– Uniform Resource Locator
VGA	– Video Graphics Array
VNC	– Virtual Network Computing
WDM	– Windows Driver Model
WIL/MA	– Wireless Interactive Lectures / Mannheim
WLAN	– Wireless Local Area Network
XML	– Extensible Markup Language
YUV	– YUV color model (Y=Luminance, U,V=Chrominance components)

1. Introduction

Lectures nowadays deal with complex topics which students have to learn. As there are different ways of learning, e.g., taking notes, repeating, etc., it is worth supporting the learner in as many ways as possible. Students' common strategies in lectures are to take notes or to write down the complete lecture making use of shorthand notation. As there are more and more electronic media in lectures, it is increasingly hard to keep up with the proceeding speed of the lecturer in handwriting. Especially when a student tries to follow an intricate piece of thought, it is possible that he or she misses the next point. Thus a lot of students wish to have a recording in order to be able to only re-play the crucial parts.

Meanwhile many Learning Management Systems (LMS) are available which enable students to use electronic media independent of location and time. Nevertheless, lectures are one of the most common ways to teach groups. In order to integrate the content of lectures into an LMS, the most evident way is to record them. Therefore, it is no big surprise that lecture recordings are often used for repetition and exam preparation.

1.1. Basic Idea

Lecture recordings have become rather popular in recent years because they are easy to achieve (Lauer & Ottmann., 2002). They stay easily achievable as long as certain constraints are taken into account, such as the availability of specially equipped lecture halls, recording equipment, additional manpower, and additional financial means.

Such restraints being quite common, there are different approaches to realize lecture recordings, e.g., completely software-based systems like the screen recording software "Camtasia" (Camtasia, 2009) or mixed hardware and software systems (Ma *et al.*, 2003). Both systems are limited concerning the different media used in a lecture. The current basic-level software records the lecturer's slides and the spoken audio. There are two reasons for this: first, it is relatively easy to record exactly these two parts which typically contain the most important information of the lecture. Second, an extra effort would be needed to record a video of the lecturer, to record audio and video of questioners, and other details which are part of the experience of a real lecture: this additional effort should not have to lead to an extra cognitive load for the lecturer.

It is important to reach a certain level of recording quality to ensure that the recording is understandable. This is mainly true for the audio track but also for the visual track. The minimum quality generally accepted by an audience can be described as telephone quality for audio and as surveillance quality for video. Nevertheless, today's TV quality has raised our expectations: it sets the standard to a HiFi sound experience and to the typical Standard Definition (SD) TV resolution as a minimum. From the cinematographic point of view, events have to stay interesting even if the spectator is at a remote location and/or if the transmission is broadcasted later. One can think of TV as the form of producing video with the best possible quality.

At the other end of the scale, there are recordings of events taken by inexperienced people or by a surveillance video system. The quality achieved by a recording is in direct proportion to the effort put into it. Of course, it is usually impossible to hire a complete camera team, well-trained and experienced, for every single lecture. Even renting professional equipment to ensure a good technical quality is expensive. The benefit of having perfectly recorded lectures often does not justify this effort. Consequently, lecture recordings are typically done by teaching assistants, which is not their main job. Their goal is to generate a "standard" recording as fast as possible and with a minimum amount of work. A specific training in composing or editing the scenes of the video can not be expected.

As a result, we often experience that these lecture recordings do not meet the level of quality one is accustomed to. While investing money into professional equipment is one part, there remains the problem of recording a lecture like a professional camera team without spending money continuously. Typically, the recording systems used in practice produce very static and hard-to-follow recordings, completely independent of how exciting the original lecture was. This is a serious deficit for e-learning today.

As a consequence, in this dissertation, we propose a lecture recording system that is able to control several cameras and audio streams automatically, imitating a professional camera team.

1.2. Starting Point

In the following sub-chapter, we will take a short glimpse at the current situation of lectures, e.g., types of media and equipment used by the lecturer and the students. In

addition, we ask which materials are used by students for exam preparation. From there we proceed to possible improvements.

1.2.1. Traditional Lectures Today

In today's lectures, a large number of lecturers do not use transparencies or slides any longer but their electronic equivalents, e.g., PowerPoint slides directly out of a computer using a projector. In fact, many students have also modernized their tools. Printouts disappear gradually and are replaced by notebooks or netbooks. Some students even use Tablet-PCs, supporting their habit of taking notes directly on the printouts or on their electronic equivalents. This fact favors the typical "lean back" situation of students in lectures: they simply want to consume the lecturer's input while participating actively and asking questions is unusual. However, these modern techniques have many advantages: it is very easy for students to download the material of the course and exercises are sent back via e-mail or by uploading them into an LMS, for example.

In order to break up the lean back situation in lectures, Scheele (Scheele *et al.*, 2003) developed the "Wireless Interactive Lectures / Mannheim" (WIL/MA) toolkit enabling students to ask questions using a built-in chat software to give feedback to the lecturer whether to speed up or to slow down the lecture, and which is most important, an online quiz tool allows to run interactive quiz rounds during a lecture. This is made possible by adapting the software to mobile devices (see Scheele *et al.*, 2004). The lecturer takes a set of questions out from the question pool of WIL/MA and starts the quiz round. Typically, two to four questions have to be answered in one round and in a limited time, e.g., five minutes. The questions are displayed on the lecturer's machine, therefore also on the projector, and in addition they are transmitted to the Personal Digital Assistants (PDAs) handed out to the students at the beginning of each lecture. Every single student is able to answer the quiz questions personally and send his or her answers back. The answers can be received in an anonymized or a personalized version, depending on the lecturer's and student's preferences and on their agreement. In case that the answers are personalized, each student can overlook his or her own development over time. When a quiz round is finished, the cumulated results are displayed on the lecturer's computer and on the projector while the individual results are sent back to each student's PDA. This provides two direct advantages: the lecturer

gets a feedback from all the students and knows whether to elaborate a topic and each student is able to compare his or her answers during the lecturer's discussion and explanation of the solutions. It is obvious that using this tool does really break up the lean back attitude of students.

The next important step of a student's participation in a lecture is his or her preparation for the exams at the end of a term. Again, it is well appreciated by students to have easy access to all the provided materials of a lecture and the associated exercises by simply downloading them from the Web or out of a Learning Management System. But, do they really have all materials accessible for download? Most of the lecturer's explanations given during the lecture have been taken down only in the form of notes, and it is very likely that some facts or at least some essential details will be missing. Furthermore, in case of a student's absence from one or several lectures, for example due to health reasons, part time work, or remote studying facilities, this additional information is missing. This includes not only the lecturer's explanations but also all questions asked during the lecture by fellow students.

It is well known that there are different ways of learning. Some people learn by just listening to explanations, others have to read critical parts to remember them, the next group memorizes facts by writing them down, even repeatedly, and some can keep things in their mind best when reproducing and practicing them. The typical lecture supports those learning best from reading materials and by reproducing exercises. Others, learning best by re-writing, may start to copy books or other written materials manually but the lecturer's explanations are lost for them at least partially. Those preferring spoken words to get the real message must attend a lecture. As lecture recordings by students are forbidden in many countries, the students do not have the possibility to produce a recording and replay certain parts of a lecture in their preferred medium.

1.2.2. Room for Improvement

Lecture recordings have filled the space to provide another medium for learners. It developed over time from simply putting a camera in front of the lecturer, adding microphones, employing staff for recording, and finally building fully-equipped multimedia lecture halls while constantly improving the achievable technical quality. As one consequence, it became much easier to produce lecture recordings. First, the re-

cordings were taken on video tapes which over time yielded to digital storage on computers. Thus, the ground was prepared to integrate lecture recordings into an LMS. The next step took pedagogical considerations into account to break up recordings into units, e.g., based on sub-chapters. This led to a natural index with smaller pieces of recording which are easier to remember: smaller units can also be re-used more easily. The re-use of learning units is an important topic nowadays (Rensing *et al.*, 2008).

Generally spoken, the focus in current research leads away from producing lecture recordings towards their immediate usage. In the next chapter, we will have a closer look at boundaries which still give room for improvement in producing lecture recordings.

1.3. Constraints

Producing lecture recordings in a well-equipped environment normally involves quite some staff to operate the electronic equipment. As it is normally not feasible to employ a large number of new people to record lectures for financial reasons, staff on hand in the department should be able to run the recording. In most cases, these persons are not skilled to fulfill this task. This often leads to failures during recordings, e.g., a missing audio track, at least at the beginning of a recording. There are two ways to handle this problem. It is possible to either accept those failures or to try avoiding them. Avoiding means that staff has to be trained to the equipment used. Doing so gets more complex the more different and complex equipment is employed. Therefore the answer to this challenge varies from recording the slides and the lecturer's audio over an additional talking-head video of the lecturer to really important lessons where a professional camera team is hired, always depending on the local circumstances.

The typical constraints of the different ways to record a lecture are discussed in detail in the following subsections.

1.3.1. Types of Recording

There are different types of recordings that have been used for lectures. The best way to differentiate is to name them according to their origins:

- surveillance recording
- meeting recording
- documentary recording
- presentation recording

While *surveillance video recording* gives the best overview of the whole scene, it often lacks details. On one hand it is important to provide an overview in order to enable the audience to be oriented; on the other hand, details are necessary to follow a lecture or to read the slides. Therefore, a lecture recording system should include both: the possibility to get an overview of the classroom and detailed shots as well.

Meeting recordings are often made by using multiple cameras or using a 360-degree camera. A computer-based screen, e.g., an electronic white-board, may also be available as an additional video stream. In this scenario, the visible camera is often directed at the speaking person or at a person determined by the moderator of the meeting. The captured computer screen is manually switched on when necessary.

In case that a 360-degree camera is used, the recorded video shows all participants sitting at a table in one video but it delivers a distorted image at a very low resolution which has to be deskewed first. Out of this processed image, the speaking person can be shown in detail, similar to the multiple-camera setup but with a much smaller resolution and therefore with fewer details. All in all, a meeting recording is an opposite of a surveillance recording. It delivers many details, for example the mimic of a participant or a clearly captured computer screen, but does not give a good overview of the entire scene.

Documentary recording is normally used for feature films or documentaries. After recording all parts scene by scene the material is edited in a post-production. It is not intended for live recordings but gives the best trade-off between giving an overview and presenting the important details. In addition, cinematographic rules are taken into account which makes the result much more interesting. It is the most complex and most expensive way of recording, and has been done successfully for years for the “Telekolleg” by the Bavarian Television in Germany (Telekolleg, 2009), for example.

At first glance, *presentation recording* comes really close to lecture recording but there still is a difference. In a presentation, it is only important to get the message out of the slides and out of the audio of the presenter. An overview of the whole scene or a shot of the live audience is therefore not necessary. Everything is recorded live and without any disturbance of the presenter. In the best case, no post-production is necessary, and all the important contents have been recorded at once. There have been some approaches to record presentations using a high-definition overview camera, either by steering a Pan-Tilt-Zoom (PTZ) camera following the presenter or by cutting out a pixel set in standard definition showing the presenter. Another way of following him or her is to evaluate the image of a PTZ camera, trying to determine where the presenter is and following her or him by steering the camera according to the detected motion direction. Even arrays of microphones have been used to determine the presenter's exact position and to steer a camera accordingly.

Finally, let us focus on the parts of the considered types of recording should be supported by lecture recording. At first, the recording should be as unintrusive as possible for the lecturer and the students. Second, it should be recorded live in order to ensure a quick availability for remote students and/or to make live streaming possible. Third, the minimum to get recorded are the lecturer's slides and the audio. Fourth, a camera recording of the talking head is desirable in order to record gestures, corroborating the lecturer's explanations. Fifth, as lecture halls and fellow students vary from lecture to lecture, an overview shot as well as a shot of the live audience helps the remote audience to get oriented. Sixth, details should be shown when necessary, and the live audience or the overview of the classroom can be shown if there is space. Seventh, in case of questions being asked, the question and if possible the questioner should be recorded as well. Last but not least, cinematographic rules such as the maximum duration of each shot should be taken into account in order to provide a non-boring version of the lecture.

In short, as many advantages of the earlier types of recordings should be combined and used for lecture recordings. It is a real challenge to consider all these features.

1.3.2. Aesthetic Considerations

The main difference between attending the live audience of a television production and watching it on television consists of two points: first, the atmosphere of the live audience, and second, the personal point of view. On the one hand, the spectator is fixed to his or her chair and is only able to turn his or her head to follow the action, but is fascinated by the created atmosphere. On the other hand, there is a lack of the live atmosphere at home. However, in order to compensate for this, the remote spectator gets new visual stimuli again and again from the different points of view of the cameras. Important details are focused and turns are taken with long shots, providing a good overview of the whole scene.

Let us solely think of a television production in which a camera is mounted on a chair with a perfect view of the set. Maybe the camera is able to pan and tilt but cannot zoom in. Watching this production on television will be boring after a short time, completely independent of how fascinating the original event is.

The usage of multiple cameras for professional productions leads to a more complex scenario. All cameras have to be coordinated, and the director decides which of them is to be on air. It is important which shots are shown in what sequence. If a wrong combination is chosen, the spectator in front of his or her TV set gets confused. There are several mistakes likely, e.g., the audio track differs significantly from the camera aperture in the video track: two people in a dialog and shown alternately seem to look in the same direction; a shot containing many details is shown for a very short time, etc. These are only two examples of mistakes which should be avoided by well-defined rules developed from cinematographers, cameramen, and directors over years. These *cinematographic rules* are an important basis which every cameraman and director has to rely on from the beginning and which has to be improved over and over again.

If these rules are neglected, the spectator gets confused, irritated, maybe disappointed, but at least distracted from the message of the production. For “light programs”, this is bad enough, but for lecture recordings which shall provide students with another learning medium, it is quite inadequate. Therefore, cinematographic rules should be taken into account for lecture recordings just as they are for live television productions.

1.3.3. Financial Constraints

A well-equipped large lecture hall normally has a sound system for local amplification and for recording in parallel, a video recording system, lighting equipment as well as a central processing and controlling unit. Depending on the number of people operating the equipment, an additional intercom system is necessary for communication during the recording. Purchasing all this equipment requires an investment of at least 20,000.00 €. This sum is easily reached by a video mixing console, including control video monitors, intercom and tally connectors, i.e. the red light showing which camera is on air, (e.g., about 11,500.00 € net price), a digital audio mixing console (e.g., about 2,100.00 € net price), and a broadcast camera set including camera, tripod, camera plate, battery pack, bag, and remote unit (e.g., about 10,300.00 € net price each). Traditionally, a larger amount is needed in the course of time for recurring expenses, e.g., the labor costs of employees necessary for operation and maintenance of the equipment.

In order to get an idea of the number of people involved in such a scenario, one must have a closer look at the details of the different systems. For video recording and processing, there is one cameraman per camera, e.g., one for the slides, one for the talking head, and a director. For audio recording, there is one audio engineer needed, and for the lighting equipment another one. In case of an important session, an additional expert is needed to supervise the recording at the central processing unit. Thus, even for this small setup, up to six people are needed. This number can increase easily as the number of cameras increases, assistants for the audio recording may be necessary, and additional lighting operators may be added.

In order to specify the cost in more detail, a snippet of an invoice for a live production is shown in Figure 1. This snippet is based on real prices, but has been anonymized; its items were split up between staff costs and equipment rental costs. This invoice gives a solid base to extrapolate potential costs for our scenario.

Description	Quantity	Price per piece	Sum of position
1 Day for setup and cutback:			
Staff	4	120,00 €	480,00 €
1 Day of shooting incl. equipment and staff:			
Camera operator	4	250,00 €	1.000,00 €
Live editor	1	260,00 €	260,00 €
Audio engineer	1	250,00 €	250,00 €
Assistants to the live editor and to the audio engineer	2	170,00 €	340,00 €
Stream operator (Final signal verification)	1	200,00 €	200,00 €
Camera, wide angle lens, tripod	4	210,00 €	840,00 €
Digital video mixing console incl. 4 screen bridge	1	120,00 €	120,00 €
Digital audio mixing console 16 channels	1	80,00 €	80,00 €
Videoscreens for controlling	1	70,00 €	70,00 €
Wireless microphones	6	60,00 €	360,00 €
Radio based intercom	6	40,00 €	240,00 €
Stream encoder (incl. fallback)	2	150,00 €	300,00 €
50m AV cables (Audio, Video, Tally)	2	80,00 €	160,00 €
Roadway safety / cable guides	1	300,00 €	300,00 €
Insurance of the equipment	1	400,00 €	400,00 €
Travel costs:			
2x mileage head office - location & return	440	0,35 €	154,00 €
less 50 km included mileage	50	-0,35 €	-17,50 €
		Net Sum	5.536,50 €
		19 % VAT	1.051,94 €
		Gross Sum	6.588,44 €

Figure 1: Invoice snippet of a live production

It includes four cameramen, a director, a sound engineer, a stream operator, and two assistants. This is almost the team size we will refer to later on.

1.4. Related Work

Many projects on the video recording of lectures have been carried out over the years which come more or less close to our system. During our research, we have found many examples in which a project borrowed an idea of a neighboring area and put it into a new context. It is therefore necessary to take even those areas of video recording into account which do not have an obvious connection to our purpose. Overall, we give a short overview over work on video recording done by researchers.

We present related work in six categories:

- Surveillance recording,
- Meeting recording,
- Documentary recording,
- Presentation recording,
- Lecture recording, and
- Additional related work.

All these works are in more or less close relation to our work as they focus, e.g., on tracking people using PTZ cameras, following a person using multiple cameras, try to react in a way on events of the environment, like speaking participants, provide video based learning materials employing a human camera team, describe details which are of use in presentation or lecture recording, how cinematographic rules can be analyzed, implemented and applied, etc. Finally, the additional related work sub-chapter presents work, which is related to our system but can not be categorized into one of the above. Occasionally during all these sub-chapters we mention details to present the relation or the difference between their work and ours.

1.4.1. Surveillance Recording

Surveillance recording is used for the conservation of evidences although it is necessary that all detected events can be evaluated retrospectively at least in the context of unattended surveillance. Of course, the major advantage of unattended surveillance is the reduction of salaries. Instead of observers surveilling the monitors, an automated system checks for suspicious events and informs a human supervisor who may be at a remote place. By doing this, a smaller number of employees have to be paid, and many different locations can be combined at one supervising station. Modern surveillance recording systems are not only able to detect suspicious events but they are also able to trace persons even across multiple cameras. The human supervisor sees the traced track marked with a certain color when investigating the recordings.

Some of the surveillance recording projects are related to our work, e.g., in case of PTZ-cameras and automatic person tracking system to steer them. A good example for this type of work is (Hampapur *et al.*, 2005) which uses multiple cameras, including PTZ-cameras, to track objects in a 3D-virtual world.

At Cornell University (Mukhopadhyay & Smith, 1999) use a two-camera system to index recordings by recognizing the transitions from slide to slide. There is an overview camera showing the lecture hall and providing the signals for synchronizing the slide transitions for the post-production. The second camera uses its hardware built-in person tracking algorithm to follow the speaker. Both camera types and perspectives come directly from surveillance video recording. Even if the system is used for presentation or lecture recording in the end, the look-and-feel of the videos is strictly that of a surveillance video. Therefore, we put it into this category.

Already here the combination of multiple cameras and an overview shot is given, but as we want to record lectures including as many details as possible but do not only want to record overview shots or persons being tracked, it is obvious that this type of recording differs significantly from our approach and task.

1.4.2. Meeting Recording

Compared to surveillance videos, meeting recordings are much closer to our main focus of lecture recording. Nevertheless, some basic ideas can be transferred: Tracking a person in a meeting room is the most obvious commonality, and it is followed by the idea of detecting events and/or faces to trigger certain behaviors of the system. If a participant of the meeting has to look after the recording, he or she may get distracted from the meeting topics. If another person has to provide this additional service, he or she has to be employed.

The usage of multiple cameras can be found very often, but even in this case a reduction to the necessary minimum can be observed. While a room can be equipped with several cameras just for the overview and some for more detailed shots, the use of PTZ cameras and/or 360° cameras can reduce this “battle of material”. As a meeting’s number of participants can widely vary from only two people up to a large group, different approaches have been considered.

(Rui, Gupta & Cadiz, 2001) used a 360° camera to capture all participants and provide the overview shot showing all participants as well as cut out images showing one single person at one time. They compared different recording modes, with and without the overview shot, let a computer or the user decide which person to show, e.g., to show the speaking person. This approach was an important contribution to improve the way meetings are recorded as it provides a reaction on the environment.

(Cutler *et al.*, 2002) amended the meeting recording with additional cameras for different views. Besides the 360° camera, they use a white-board camera and an overview camera for the meeting room. To track the speaker they use a microphone array. They have implemented a simple version of a virtual director module deciding which view or part of view will be shown. It directly depends on the result of the sound source localization routine of the microphone array and on the amount of motion in the different video sources. In order to not switch too often between two shots in a

discussion, the director tries to choose a large shot showing both speakers side by side which may be possible with a 360° camera. While this approach already employs multiple camera views to give the spectator a more global impression of the scene, it is not yet coordinated by cinematographic rules.

Earlier, Siemens Corporate Research employed a 360° camera approach in larger rooms as described by (Huang, Cui & Samarasekera, 1998). They put considerable effort into tracking multiple people in a room by multiple sensors and cut the corresponding parts out of the 360° image. Nevertheless, the resulting video is something in between a surveillance video and a meeting recording. Here, the idea of showing the relevant person most of the time is coming up and presenting a global view is implemented.

1.4.3. Documentary Recording

Documentary Recording is normally done by human camera teams; it does not have much to do with our approach. Nevertheless, as even documentary recording is used in distance learning scenarios, it is worth having a short glimpse at this type of video.

In order to employ good learning materials for lecture topics, those institutions having access to professional camera teams are able to produce documentary recordings comparable to a real movie which is really expensive. While human camera teams perfectly know how to produce the video, the dramaturgy must be appropriate to transport the content to learn without distracting the learners. Good examples are the (Telekolleg, 2009) from the Bayerischer Rundfunk (BR) in television, the (Funkkolleg, 2009) from the second radio program of the Hessischer Rundfunk (HR), and the videos shown on the open2.net-portal (Open2.net, 2009) providing the students of the United Kingdom Open University with video learning materials produced by the British Broadcasting Corporation (BBC) in the Internet. These works make obvious that producing learning materials out of video presentations and their recordings is not an easy job and even true professionals need time to achieve the renowned quality.

For purely academic purposes, (Rößling & Ackermann, 2007) presented a framework for generating AV content out of formulas and/or algorithms. Thus, pre-produced clips dealing with topic details are available and can be presented live as well as get cut into the recording. It is a typical way in TV documentary recording to use such clips either pre-produced or generated on the fly to explain complex facts. It is a good

way of making complex conjunctions visible, which is necessary for the recording of learning materials.

1.4.4. Presentation Recording

Presentation Recording comes very close to the target we are aiming at. The main difference is its purpose: While lecture recording aims at transferring knowledge to the audience and enables people to recapitulate and to prepare for examinations, presentation recording main purpose is to promote something, e.g., current research results, new prototypes, how to apply new products correctly. Therefore, a special didactic preparation of the recorded content is not compulsory but can be helpful. We differentiate these two types of recordings by looking at their context. If they explicitly focus on lectures, they will be discussed in the lecture recording section, otherwise, they are discussed here.

Early approaches of presentation recording mainly focused on the technical part of the job. (Cruz & Hill, 1994) recorded the presenter, as well as his or her audio and the slides. After synchronization the media sources are presented together on one screen, switching the slides at the recorded points in time. In contrast, (Bianchi, 1998) mainly tracked the presenter automatically and switched between multiple cameras based on the action shown in the images. As many presenters visually refer to their slides during their presentation, the main points are shown in most cases. Bianchi also mentioned that implementing cinematographic rules in his system may be a useful future work. Nevertheless, in (Bianchi, 2004), he re-presents his system together with some of his experiences: while cinematographic rules do not seem to have been implemented. Amongst other reasons, these papers encouraged us trying to implement cinematographic rules in automated presentation or lecture recording systems.

(He, Grudin & Gupta, 2000) asked whether presentations should be designed in a special way for on-demand viewing. They conclude that, as presentations are already well structured, it is not necessary to prepare them for later on-demand viewing but it can help to optimize the presentation. For presentations as well as for recordings done by professionals the base is the idea of a storyboard, which is also the base for cinematographic rules.

The development of presentation recordings continued for example with (Baecker, 2003). He intended to provide a local audience as well as a remote one with the recording, and his main focus was set on scalable video streams in order to enable a large variety of remote audience. As the camera and directing work was done by a human camera team, cinematographic rules have been used but it was clearly an expensive project. So, they tinkered with the idea of automating the recording, which gave us another good reason for starting our work.

Interestingly, Rowe and Casalaina stated in their paper (Rowe & Casalaina, 2006) that it is possible to capture conference and workshop presentations for \$3,000 to \$5,000 per day employing a real crew and standard equipment. They use a so-called straight-to-disk strategy in which most of the post-production and its costs are eliminated. They state that recording a session at this price is feasible even for the limited budgets of conferences and therefore automatic recording would not be necessary. This conclusion is curious as they describe the equipment and the techniques in every detail but admit that at the same time some details of the recording have to be improved in the future. These details are, e.g., that they want to use more than one wireless microphone for the different speakers, an additional camera for the audience if no one objects, and pan-tilt-zoom cameras to give the director more control. All these details would have been taken into account beforehand by a professional human camera team which is of course more expensive. This paper makes obvious that a trade-off between complexity of the system and the costs for it and its operating staff is necessary. Thus, our system focuses on feasible costs while having a complex distributed system which tries to minimize the user's interaction.

Rowe also did research on automatic presentation recording, for example, in (Machnicki & Rowe, 2002) in which basic cinematographic rules have been realized by hard-coded nested if-then clauses and the detection of questions out of the audience was done by room microphones. Here we found the complexity of reactions on the environment and the need for good sensors and a high audio quality.

A similar approach was chosen by (Rui, Gupta & Grudin, 2003). They determined and described useful cinematographic rules and checked which of them are feasible. In the first version of their system they determined basic cinematographic rules and implemented them. Finally its results were compared to the results of professional

videographers by representative viewers as well as by the videographers in order to get it statistically evaluated. It turned out that the representative viewers judged both results overall nearly equal, while the automatic system got slightly lower values in detailed issues resulting in an overall quality value slightly below the professionals video. One disadvantage stays, i.e., the predictable behavior how the virtual director does a sequence of shots. This results from the hard-coded rules with fixed weights used for the director's Finite State Machine. This project and its results are therefore closely related to our work, even if we do not use hard-coded rules in our virtual director.

In 2002, the FX Palo Alto Labs of Xerox developed the FLYSPEC system described in detail in (Liu *et al.*, 2002a) as a remote inspection system. While a computer proposes one shot multiple users can simultaneous request different shots. Its improvement concerning different simultaneous video requests was presented in (Liu *et al.*, 2002b). As the resulting videos are still recordings of a presentation, even in case that there are many different videos of one presentation, we listed this work in this category. The system consists of a high-resolution camera used for an overview shot, for the tracking of the presenter, and for cutting out of images in standard definition (SD). The second camera is a PTZ camera. It is controlled by the routines having the input from the high-resolution camera as well as by the commands of multiple users demanding a certain view. Depending on these requests, the system chooses whether the demand can be fulfilled by the PTZ camera or by a cutout of the high-resolution camera. Thus, virtual cameras can be interpolated out of this system. The algorithms, how to react to the demands and how to blend and generate virtual cameras are improved in the second paper. Proposing one computer's cut and enabling live spectators to create their own director's cut is an interesting approach. But as we want to steer the cameras and do the director's work based only on cinematographic rules, we do not provide such possibilities.

The areas of application are manifold. The multimedia live webcast of the Open University's worldwide virtual degree ceremony (Scott & Mason, 2001) is a perfect example case for automatic presentation recording.

1.4.5. Lecture Recording

(Truong, Abowd & Brotherton, 2001) present an overview of the different devices and applications for the automatic recording of live experiences developed at Georgia Institute of Technology. They sum up their experience in order to achieve a reference point for future developers. However, they focus on the technical part of the job and do not consider the way how cinematographic rules could improve the recorded videos. Even more they tried to use statistics to discover design rules for lecture recording which leads into the danger of uniform approaches.

The basic version of lecture recording consists of a recording of the slides, the lecturers' audio, the audio of simulations and animations, and the synchronization of the recorded streams. In many cases, the recording gets manually indexed in order to find a specific part of the content more easily. Sometimes, a video of the lecturer is recorded additionally, a so-called "talking-head" video. All these approaches use either a browser or incorporate a browser-component in their software to show the different recorded streams arranged on one display. Even if every researcher focuses on different sub-tasks, at a first glance it seems like much of the work has been done more than once.

A typical example of the standard recording is found in (Brotherton, 2001). He takes the slides from capturing white-board software, records a talking-head video, includes audio and syncs, and manually indexes them in a post-production step. Similar to this is the system of (Dal Lago *et al.*, 2002). The main difference is that they use two analog videos, one for the talking head and one for the slides. Both videos get converted into audio-video files and again get synced and indexed in manual post-production. These manual post-production steps are a key cost factor which we try to avoid.

A little simpler is the recording system of (He & Zhang, 2007). They record only the white-board and the lecturer in front of it using one video camera as they have remote collaboration in mind. It is obvious that the origin of their lecture recording lies in the appearance of specially equipped multimedia lecture halls. The advantage of presenting multiple media integrated into one lecture hall awoke the desire of recording them. Anyhow, in most descriptions of multimedia lecture halls, recording only plays an inferior role and is therefore only mentioned on the brink, like in (Mühlhäuser, 2005) and (Rößling *et al.*, 2006). Depending on the recording setup it is possible to

get into trouble with the lighting conditions which are different for the white-board and the lecturer in front of it. By using different video sources our system avoids such problems. A typical representative for the first step in lecture recording is described in (Rowe *et al.*, 2003). Here, expensive staff remains necessary, it should be reduced over time by implementing more intelligent software. It already uses separate streams but instead of automating the final cut by employing a virtual director still a lot of manual work is necessary.

At the University of Freiburg under the direction of Professor Ottmann, a lecture authoring tool was developed, beginning in 1996. It is meanwhile shipped as a product by the spin-off company “imc AG”. During the initial research and implementation phase, the working title was “Authoring on the Fly”; the product is now called “Lecturnity”. The two programs share the same basic approach but Lecturnity was improved over the years and amended with many details, simplifying access and usage of the recorded material. The progress of the development is presented in the papers of (Datta & Ottmann, 2001) which handle the use, the possibilities, and the implications of multimedia enhancements in offline and classroom lectures. (Hürst *et al.*, 2001) focuses on the human computer interfaces for the lecturer at recording time and for the user for replaying, (Hürst, Müller & Ottmann, 2004) address the automatic production of multimedia material for teaching purposes, for example. It was soon clear that this type of creating learning materials provides many advantages, e.g., it is relatively cheap to produce, it is fast to achieve, it is easy to automatically structure the content, and it features the universities’ professors and their specialties which was presented in detail by (Lauer & Ottmann, 2002) and by (Müller, Ottmann & Zhang, 2002). While this project focuses on the production of learning materials for LMS which leads to a certain amount of complexity, we focus on lecture recording including an easy access to the content for the students.

Indexing during post-production makes the recordings easier to access but, as long as a computer is not able to fully understand the semantic content of a scene, much work has to be done manually, as (Liu & Kender, 2004) stated. (Müller & Ottmann, 2000) proposed a way of preparing and doing recordings robustly for indexing, and (Wang, Ngo & Pong, 2003) focused on a two step approach for syncing video frames to electronic slides. At first they use text recognition in videos based on multi-frame integration in order to achieve the binarized text and in the second step they separately com-

pare the detected titles and the detected content with their pendants written on the slides. In this paper, they achieve an accuracy of matched slides from 82.4 % to 92 %. As our entire lecture recordings are separated by the sub-chapters of the lecture, we decided to keep this simple but very effective way of indexing.

Another important task is to keep the recording easy. (Mertens & Rolf, 2003) tried to extract characteristic parameters and features in order to find the ideal lecture recording tool and set up their “Flying Classroom”. Their work is very similar to the “teleTASK” system built under the supervision of Professor Meinel at the University of Potsdam, described e.g., in (Ma *et al.*, 2003). The third representative of this type is described by (Shi *et al.*, 2003) in their “SmartClassroom” project. However, those systems tend to hold the lecturer captive in a virtual cage formed by the section the image of the lecturer camera shows. We decided that such limitations are not wanted for our system. The use of multiple and/or PTZ cameras can damp this effect but will not remove it completely, as done by (Gleicher, Heck & Wallick, 2002). This team modified their system in order to by now crop an SD image out of a high resolution image in post-production then applying cinematographic rules in order to generate new image arrangements and video transition effects between shots, as described in (Heck, Wallick & Gleicher, 2007). Trying to compensate different camera positions by zooming is a very unnatural way as a human being is not able to zoom in or out of his or her view. Therefore, we decided to keep our approach of different camera positions and a virtual director.

Another demand is to focus on the lecturer’s workload and therefore to record “lecturer oriented” as (Häussge *et al.*, 2008) state. Typical examples for approaches focusing on simple usage are the following two papers. (Ziewer, 2007) does only record the content and the lecturers’ audio using Virtual Network Computing (VNC), based on the recording features of software like UltraVNC and TightVNC normally used for remote desktop access scenarios. (Yokoi & Fujiyoshi, 2005) use the idea of cropping a standard definition image out of a high-resolution one in order to be able to compensate the lecturers’ movement. (Zhang *et al.*, 2005) amend the virtual tracking inside a high resolution image by using a PTZ camera, leading to smoother camera movements. These papers showed us the importance of taking care of the lecturers’ workload.

Another important project bridging the gap between multimedia lecture halls and automatic lecture recording is the E-Chalk project of the Technical University Berlin, well described in (Friedland & Pauls, 2005) and (Friedland, 2006). It focuses on high-quality recordings of the electronic board and of the audio. Playback is possible on many different devices, including mobile phones. Its focus is on high quality recordings of the different AV sources, but does not use any cinematographic rules.

However, in most cases, researchers do still focus more on technical details than on how to improve the *impression* the recording has on its spectators. At first, multiple cameras are necessary in order to have some recording options and then the choice of the camera must be done carefully. We claim that cinematographic rules are necessary in order to create a lively and vivid experience for the learner, even if they are hard to describe to computers and hard to implement. Any such rule implemented is a benefit for the spectator.

(Onishi & Fukunaga, 2004) use three cameras showing the same scene, mainly the chalkboard, out of different angles, to optimize the framing of the image, i.e., selecting the angle in which the lecturer does hide the fewest amount of information on the board, based on the lecturer's motion but without the use of cinematographic rules. (Hartle *et al.*, 2005) use multiple perspectives but takes advantage of basic cinematographic rules for the arrangement of shots only and does not use a virtual director.

Microsoft Research runs a project called iCam/iCam2 which uses a virtual director based on a Finite State Machine. It partly uses cinematographic rules as all commands are written in pseudo code including *if-then* clauses. The commands itself are brief tokens describing literally the basic actions to perform. The development started with some prerequisites concerning the camera management in (Liu *et al.*, 2001), leading to their first version presented in (Rui *et al.*, 2001). In 2004, they presented the improved version (Rui *et al.*, 2004), followed by a portable version presented in (Wallick, Rui & He, 2004). The newest version called iCam2 was presented by (Zhang *et al.*, 2008). The system was improved in capturing computer-based visuals (animations, simulations) using a capture card, employing a microphone array for questions from the audience, reducing the amount of cameras to track the speaker, and finally developing a scripting language to replace the note form of cinematographic rules. As these rules are nevertheless hard-coded they lead to a predictable behavior of the vir-

tual director and might distract spectators from the content. As we do not use hard-coded rules for our FSM-based virtual director, this point is the main difference to our system. While at a first glance this project seems to be similar to ours, it focuses mainly on the development of the technical solutions for recording and transmitting while we focus on the implementation of more and more complex cinematographic rules.

1.4.6. Additional Related Work

Besides the related work concerning the major goal of recording live events, there are some other topics worth having a closer look at as they provide important details. Some of them deal with background information gathering and some of them deal with basic research e.g., on cinematographic rules and their different implementation approaches; on *interactive* lectures setting the base for our system; on improving the sound quality of audio recordings in lecture halls; on positioning techniques for people inside lecture halls; on additional sensor equipment providing more intuitive human-computer interfaces (HCI).

The use of *cinematographic rules* in automatic lecture recording is a typical difference between a professional video production and a home video production. For example, these rules determine how long a certain shot should last, how to frame a person in an image, and many details more. A more detailed introduction and explanation of cinematographic rules is given in chapter two.

Cinematographic rules have been used in many different scenarios: in a virtual 3D-environment, e.g., in the camera control for camera motions (Christianson *et al.*, 1996) which focuses more on image arrangement or in virtual story telling (Courty *et al.*, 2003) where more weight is put on shots and their transitions. (Gleicher & Masanz, 2000) wanted to generate new shots, views and perspectives out of a given video image based on cinematographic rules concerning image arrangement. These papers assured us in using such rules in order to achieve an improved result as it is also an issue in virtual 3D environments and storytelling and may be helpful even in post-production.

(He, Cohen & Salesin, 1996) set up an efficient system to use cinematographic rules in a virtual world. They define very fine granular sets of camera modules and idioms describing scenes or shots and the cameras used for them. As all rules have been hard-

coded and any action to be recorded is perfectly known as it is generated by the same machine in the virtual world, it is no problem for them to react precisely and to choose the correct idiom. In contrast, in the real world, it is a big problem to reliably detect a relevant action and in order to not get easily predictable the rules should not be hard coded. This article showed us the differences between applying cinematographic rules in a fully controlled virtual environment and in the real world.

(Matsuo, Amano & Uehara, 2002) went another way: They tried to extract cinematographic rules out of given videos or movies and to apply them on new videos in order to copy the director's style of the original movie. In their work, they limited themselves to distinguish between three shot types and to shot duration statistics. Besides the fact that cinematographic rules should be applied based on events taking place and not for their own sake, there is a big variety between the different genres of movies, feature or documentary films, or videos. Even though they are all based on the same rules, the rules get interpreted differently depending on the genre. This paper shows us that statistics is useful to describe given material, but also that there are limits of extrapolation. Besides the less emotional view on cinematographic rules of (Thomson, 1993 & 1998), James Monaco gives a good impression on how various cinematographic rules can be applied and interpreted in (Monaco, 2000a & 2000b). These books were the main source of our prototype concerning cinematographic rules.

As we have seen, *interactive lectures* help to break up the students' typical "lean-back" behavior during lectures. Their development started with interactive applets in tele-teaching scenarios, as described e.g., in (Kuhmünch, 2001). A very comprehensive and extensively evaluated system was developed by (Scheele *et al.*, 2003 & 2004). It started being based on laptop computers but has also been transferred to PDAs meanwhile. Similar approaches were presented by (Choi *et al.*, 2004) and have been ported to different end devices such as mobile phones, e.g., (Bär *et al.*, 2005). Overall, interactive lectures are now well established and will be used in more and more courses in the future. The conclusion we draw is that interactive lectures need to be well supported by recording as many details as possible to keep their richness.

Audio recording in good quality is important for lecture recording as the lecturer's explanations and the audience's questions have to be easily understandable. A very good introduction can be found in the audio engineering textbook of the "Schule für

Rundfunktechnik” (SRT) respectively the “ARD.ZDF medienakademie” (Dickreiter *et al.*, 2008a & 2008b). However, it is necessary to adapt them to the lecture hall. For the E-Chalk project, (Friedland, Jantz & Knipping, 2004) concentrate on audio. In (Friedland *et al.*, 2005), the system was enhanced and it now provides a basis for our own research. Many details of these basics and experiences were used in our sound engineer module.

Indoor positioning is a technical process to find out the position of a person in a building. While for outdoor scenarios the Global Positioning System (GPS) is the state of the art for localization, GPS is not useful indoors as it is impossible to receive satellite signals. We need special algorithms for indoor positioning to be able to localize a moving lecturer on one hand and a questioner in the audience on the other hand. We discussed several ways of indoor positioning approaches. One possibility is to use microphone arrays in which the microphones are arranged, e.g., like the number five on a dice. Depending on the small runtime differences of the sound waves, it is possible to calculate the direction out of which the sound came. Details concerning this approach can be found in (Rui & Florencio, 2004). The technology was improved by the work of (Tashev & Malvar, 2005), and as mentioned above, it is now used for the iCam2 project of Microsoft Research (Zhang *et al.*, 2008). While this approach is precise concerning the direction of the speaker, there can be problems concerning the distance of the sound source to the microphone array. There also might be problems in a lecture hall if other sound sources exist, e.g., if other students are speaking at the same time.

Therefore, we decided to use another approach for localization, namely indoor positioning using 802.11 wireless LAN access points already installed at universities. The algorithms for localization were developed by Thomas King and Hendrik Lemelson at our institute. They are described in (King, Kopf & Effelsberg, 2005). Typical for this kind of indoor positioning is an average error of about 2 to 2.5 meters. We compensate this error in our system by two measures:

1. We estimate a region using WLAN indoor positioning and then let the person specify the finer granularity of his or her position.

2. We zoom not too close onto a questioner but let room for up to three seats around him or her and zoom in later, controlled by image processing in the automatic cameraman.

The WLAN indoor positioning is done using the PDAs we already have in use for the interactive lecture system (Scheele *et al.*, 2004). Here, we combined different work from our institute for new purposes. Tracking people while using this technology is not a big problem as mavericks can be excluded by plausibility checks. Thus, it is possible to track a moving lecturer or questioner. Naturally, it might be a problem to hold a PDA while moving around and maybe needing the hands free for other things.

Wearable devices are a possibility to overcome this problem. For example, the QBIC is a belt-integrated computer developed at the ETH Zürich, Switzerland (Amft *et al.*, 2004). It is a kind of PDA which is integrated into a belt; it can be equipped with various sensors (Lukowicz *et al.*, 2002). Thus, it is a feasible solution for the mentioned handling problem.

Additional sensors are able to provide more intuitive human-computer interfaces. A very good example exists in conjunction with the wearable devices as described in (Büren von, 2002). There additional special location and environment condition sensors get attached to the QBIC. One of the most prominent applications is the situation of fire-fighters exposed to hostile environments as they have to intuitively but precisely use the devices even though they wear full protective clothing. Such hardware sensors can also provide a good support for, e.g., tracking people in lecture halls. So, the idea of defining an open interface even for future sensors for our Automatic Lecture Recording systems was born.

Another possible way of improving the intuitivity of human-computer interfaces are software sensors as they are employed in image and video processing. In our case, the detection and semantic evaluation of participant's gestures in the lecture hall can help reacting on relevant actions, e.g., question announcements and when giving someone the floor. Algorithms to evaluate the semantics of movements and gestures were developed by many researchers, e.g., at our institute as described in (Kopf *et al.*, 2003) and (Kopf, Haenselmann & Effelsberg, 2004).

2. Design of the Distributed System

As already mentioned, we want to enrich lecture recordings with additional views to enable the learner to get a more complete impression of the entire lecture situation. To make the result attractive and to keep the recording vivid, we want to base our system on cinematographic rules and mimic a well-trained human camera team.

2.1. Analyzing the Real World

We first determine the “ingredients” a lecture consists of. By ranking them, we can decide the priorities for the implementation.

2.1.1. Determining the “Ingredients”

Everyone who has attended a lecture can easily determine what is needed. An inventory list would at least include

- the lecturer,
- his or her presentation slides (e.g., produced with PowerPoint),
- his or her manuscript,
- live annotations on the slides,
- animations, simulations, and video clips shown during the presentation,
- his or her spoken words,
- his or her gestures and mimics,
- the audience’s questions and comments,
- the interaction between the lecturer and the audience arising from questions.

These are the most important ingredients characterizing a lecture. We feel certain that the sum of all these ingredients is necessary to be able to achieve a complete and didactically useful recording of a lecture. For example, in addition to the lecturer’s frontal presentation, students tend to remember a humorous comment made by a fellow student. Therefore, we claim that all ingredients or at least most of them should be recorded.

It is questionable whether all ingredients are of equal importance. Naturally, there is the need for the content of the lecture at first. Therefore, the presentation slides, the script, the annotations and possibly animations, simulations, and videos shown are essential. Equally important are the lecturer’s spoken words and his or her gestures and mimics. Having determined the basic level for any lecture recording, we must

admit that this basic level is the state of the art when looking at lecture recordings today. Although lectures have so much more to offer, many ingredients are left unconsidered.

In addition to questions starting interactions between the questioner and the lecturer, interactive parts have increased in lectures in recent years in order to motivate the students and to check the learning progress more easily; these parts should be recorded as well. Let us look at an example: As mentioned above, an interactive quiz for Personal Digital Assistants (PDAs) has been developed at the University of Mannheim (Scheele *et al.*, 2004). This quiz can be used by lecturers during the lecture to get an immediate feedback from the students on how well the current topic was understood. Besides the simple recording of the questions, the statistical analysis of all given answers and the explanation of the solution, the quiz itself can be provided separately on the homepage of the course in parallel to the recording. So, remote students are able to redo the quiz on their own.

While it is rather easy to record the basic ingredients by using a simple screen recording tool, it is not possible to record the video of a lecturer and his audience in such an easy way. The screen recording tool records the presentation slides, the annotations, and the lecturer's audio. All the other parts, such as animations, simulations, videos, and all interactions between the lecturer and the audience such as quizzes can be recorded similarly as long as they are based on software running on the lecturer's computer.

It is significantly more complex to additionally record the video of a lecturer and his or her audience. In order to find a straight-forward solution, the simplest approach is to employ one camera for the lecturer and one for the audience, both used in the "very long shot" or the "extreme long shot" mode as described in (Thompson, 1998). This leads to a fully static setting in which it is very difficult to see any detail or mimic of the lecturer or to identify a student asking a question among all others. As a consequence, the lecturer's camera can be set up closer to the lecturer; then it is possible to see the desired details, but the lecturer is virtually captured in a "virtual cage" which he or she must not leave. The only way to change this is to employ a cameraman to operate the camera, zoom in and out, and follow all the movements of the lecturer. The same is true for the audience camera: While the "very long shot", mentioned

above, may be sufficient for a picture of the entire audience, it is not sufficient to frame a questioner asking a question. Everyone sitting in a lecture room will turn his or her head to take a look at the questioner, so recordings of a similar action of the camera are expected. So, we are in need of another camera operator.

However, questions need to be heard. There are different approaches to get their audio into a microphone and mix it with the lecturer's audio. Sometimes, cabled microphones are used, so the questioner has to stand up, walk to the microphone, and ask his or her question. This is a psychological barrier, so fewer questions will be asked. A slightly better idea is to use a wireless microphone and hand it over to the questioner, which works very well from an audio quality's point of view but has the disadvantage that it takes quite some time until the microphone has reached its place of action and the question can be asked. Far less intrusive is the use of an "atmosphere" microphone sensitive enough to get the question. Unfortunately, this type of microphone is omni-directional so that the fan of the video projector, for example, almost always disturbs the recording quality.

Another setup makes use of an array of unidirectional microphones which are very sensitive on only one direction. By using an array of microphones structured like the five on a dice, the direction of the audio can be calculated out of the time shift of the audio waveforms in conjunction with triangulation and can be used for position estimation along the corridor of the sensitive direction. One requirement with such an installation is the discipline of the audience. Along the sensitive corridor of the microphone, only the questioner should speak. This may not be easy during a lecture, especially if a discussion takes place. To ensure a good recording quality, it would be better if everyone had his own microphone. For such a solution, a multi-channel mixing console is necessary which leads to much higher costs.

Anyway, costs are a key factor for every solution named above. To propose a feasible approach, we assemble a useful human camera team and then focus on technical and financial constraints.

2.1.2. Details of a Real Camera Team

In live TV productions, a large number of persons are necessary to cover all parts, but for lecture recordings in universities it is neither possible nor useful to have such a large staff. For example, there is no need for make-up artists or set constructors. Furthermore, due to the relatively well-known work-flow of a lecture, a camera team would be sufficient. This is the reason why we focus on the camera team in the following.

A real camera team for a live studio production does still consist of several people. A good overview is given in chapter “Studioproduktion und Außenübertragung” of (Schult & Buchholz, 2002). As an example, we present a list of people who may belong to a team as used at Südwest Rundfunk (SWR), German television:

- director,
- editor,
- taped recordings operator,
- inserts operator,
- lighting cameraman, who is the coordinator of all cameramen and lighting technicians,
- cameramen,
- lighting technicians,
- iris operator, who centrally controls the irises of all cameras in order to achieve a homogeneous look of all images,
- audio engineer,
- and final signal controller.

Although each of them is important for a show produced at a high quality level, there are cheaper productions which try to reduce the number of people in the team. From the viewpoint of cost reduction, only a director, an editor, a lighting cameraman, and all other cameramen are necessary as a team. In smaller productions, the director and the editor may be the same person. If there is no iris operator, every cameraman has to adjust the iris on his own, which may lead to video channels using different expo-

tures. One possibility to overcome this problem is to use fixed presets, e.g., for the white balance. Though, in this case it is not possible to react to changing light environments, for example, if natural light lights the scene or the sky changes from cloudy to sunny.

In order to understand the job of cameramen correctly, we have to go into more detail. A cameraman has to work in a team, which starts from the planning phase and leads all the way through the production steps. So, the duties of cameramen are divided into three parts: a) before the show, b) during the show, and c) during a shot.

Before the show, there is a meeting of all to review the storyboard. The director goes through all the details of the show and makes clear the important points to the lighting cameraman and all other cameramen. Figure 2 shows a part out of a storyboard.

Kaffee oder Tee Fr, 02.03.07

	Pos	Startzeit	Aufz.Art	Inhalt	Ist-Länge
A	1	2	3	4 1. Zeile = SWR 2. Zeile = Kochkunststrätsel 3. Zeile = 76522 Baden-Baden	5
B	13	16:46:13	Live	Selbermachen: Dekorieren Des Sultans liebste Zwiebel <i>Wiese - Terrasse</i> <i>Pulpen</i> I.W.: ...n unserem kurzen Filmausschnitt! <i>Wintergarten</i>	06:00
C	14	16:52:13	Live	Moderation <i>in die 1</i> I.W.: ...wie in unserem kurzen Filmausschnitt!	00:20
D	15	16:52:33	DigiB	Besser leben Entrümpel dein Leben - Ordnung macht glücklich 10:08:28 - 10:09:01 I.W.: ...+M) Beitragstext/Letzte Worte: in uns	00:33
E	16	16:53:06	Live	Besser leben <i>Kü - Tisch</i> 2.3.07 Frei-Räume für die Seele I.W.: ...hts aus mit dem Entrümpeln????	06:20

Figure 2: Part of a Storyboard of "Kaffee oder Tee" of the SWR

As this is a scan from a storyboard in German language, we now describe its content in detail. Therefore, we numbered the rows from A to E and the columns from 1 to 5. Column 1 contains a sequential number, column 2 the starting time in hh:mm:ss notation. The third column describes the origin of the signal where "DigiB" stands for DigiBeta, a video recording format of Sony Corporation, and "Live" stands for live recording. In column 4, each step of the contents to broadcast is described. The last column 5 contains the duration of the step in mm:ss notation. Let us look at column 4 in more detail. It contains at first the title of the part. If its origin is a video tape, the corresponding SMPTE time code is shown additionally as in row D. The parts of the text which are written in rectangles shown in rows A and B are either used for the inserts to publish an address for a lottery in row A or to introduce a person in row B. The abbreviation "I.W.:" stands for "last words" of this part. It is the signal for the director to switch the correct signal "on air". All handwritten annotations are additional information given during the meeting by the director to enable the cameramen to do their job even better. The cameraman gets his orders in three steps: Basic information out of the storyboard, additional briefing by the director in the meeting, and live information during the show using the intercom communication system.

The next location is the studio. The position of each cameraman is crucial. As described by (Thompson, 1998), the "line of action" must never be crossed. Figures 3 and 4 show a little example from top view to clarify how important this line is:

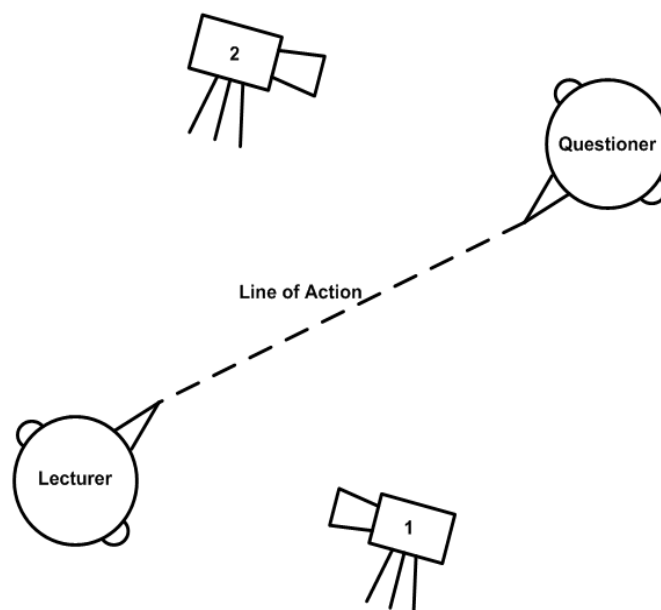


Figure 3: Example of wrong recording positions according to the "Line of Action"

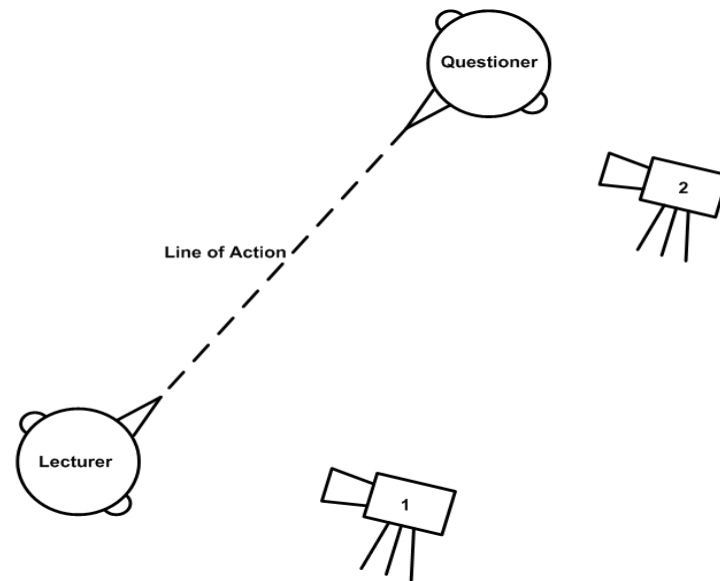


Figure 4: Example of correct recording positions according to the "Line of Action"

We argue that the lecturer and the questioner in the audience are discussing. In case A), camera 1 will show the lecturer looking from the left edge to the right and camera 2 will show the questioner also looking from the left edge to the right. If these shots are shown one after the other, the effect will be very confusing to the spectator because the two do not face each other. The reason is that both cameras are on different sides of the "line of action". In case B), camera 1 will still show the lecturer looking from the left edge to the right but camera 2 shows the questioner looking from the right to the left. Like this, the spectator gets the impression that both are facing each other while discussing. The correct place of each camera and its cameraman is very important.

We will now get into the live phase during the show. During the entire show, the cameramen use headsets to communicate with the editor in the central control room. There is only one intercom system for all participants and everyone is able to speak at the same time. Therefore, it is necessary to be extremely disciplined so that everyone is able to understand the person who is speaking.

Using the intercom, the cameraman gets his orders from the editor and the director. These orders include information about "who is on air", "who will be on air next", and "which detail or framing a certain cameraman should show". Sometimes the cameraman informs the control room, for example, if for technical reason, he is unable to perform a requested shot, or if he has an idea of an extraordinary detail or framing

which he wants to show. This conversation includes commands like: "Camera 1, please frame person A in a way that he looks from the left edge into the image." Another cameraman gets the command for the counterpart: "Camera 3, please frame person D in a way that he looks from the right into the image." Now the editor is able to switch between these two shots as long as the two people are talking to each other. For the spectator in front of the TV set, it looks as if the two are facing each other while talking, even if there are hundreds of miles between them. Through the intercom there is a continuous communication in order to optimize the aesthetic aspects of the recording.

We will now take a closer look at the shot itself in conjunction with the work of a cameraman.

At first, the cameraman has to bring the requested image into the sight of the camera by moving, panning, and tilting. Next, in case there is no iris operator, the cameraman has to control the iris himself. He constantly adjusts it in order to achieve a similarly exposed image, even if the illumination varies from one part of the studio to another. It is very important that the cameraman focuses on the main parts of the chosen image. By zooming in or out before or even while being on air, the image gets its final look. This complex process which needs a lot of experience for live productions is repeated for every single shot during the show in which each cameraman has to determine how to do the framing and the composition of each shot. While *"Framing is the process of selecting a part of a view in order to isolate it and so give it emphasis. [...] Composition is the arrangement of the objects and/or people within the frame. Its use [...] is to create the third dimension, namely depth, within the frame."* (Thompson, 1998). To select the correct part of a view at first means to catch the action in the frame, second to check whether the lighting has to be corrected, and third to check that there is no interference with the background, e.g., lines which are cutting through a head. The composition has to transport the relation of objects and/or people to each other, and it also has to support the action to make sure that the viewer is able to keep track of it easily.

If we use the procedure of live TV production directly for lecture recording without any adaptation, the team will use at least three cameras for recording all the people: The first camera will be the long-shot camera, the second will be used for details of

the lecturer, and the third records the audience and the medium-close shots of the questioners. Depending on the size of the audience or stage, additional cameras support the tasks of camera two and/or three. In order to record the slides, the script and annotation converters instead of real cameras will be used and taken as additional video sources. Including the slides, we have at least four video sources recording a lecture like a professional camera team. Figure 5 out of (Schmidt, 2005) shows an abstracted configuration of a live production in a TV studio.

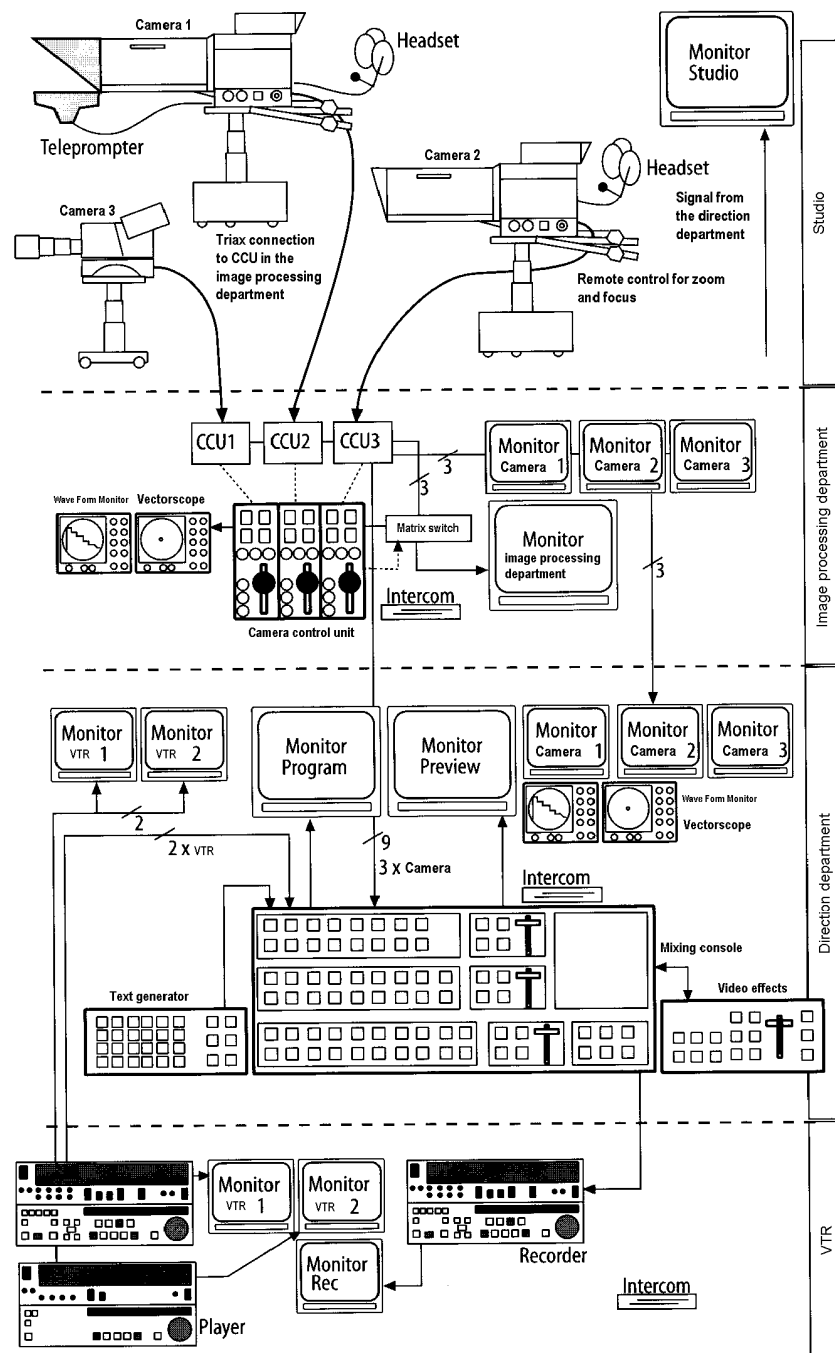


Figure 5: Schematic view of equipment for studio production (based on Schmidt, 2005)

All sources are brought together by the director who edits the sources during the production. He is in charge of the continuity of the action and its clearness to the viewer. He decides which camera is going to be "on air" next. In order to keep the action clear to the viewer, he has to follow many video production rules. Based on (Thompson, 1993) some examples of these are:

- show the action first in the larger context, then in detail;
- show a neutral scene in between to make a switch of context clear to the viewer;
- show a lecturer and questioner in discussion by alternating in a way so they are facing each other;
- show the slides in between to keep the reference to the original subject, and show the audience from time to time to show their reaction;
- the duration of a shot should be long enough so that all details shown in the scene can be perceived. Thus, the more complex a scene is the longer the duration of the shot should be. The minimum duration of a shot has to be about four seconds.

Every time a new shot appears, the recording attracts attention, and therefore the lecture recording stays interesting. However, each transition between the shots should be meaningful and not confusing to the viewer. If a viewer gets irritated, he will miss at least a part of the lecture, which makes understanding and thus learning more difficult.

Up to now the camera team consists of cameramen and a director who does only focus on the visual part. In order to record the audio, the typical setup for a TV production is to give the protagonist, in our case the lecturer, a wireless clip-on or head-worn microphone, at least one dedicated wireless microphone for questioners controlled and/or handed out by an audio assistant, and an atmosphere microphone. Sound produced by a lecturer's computer will also be taken as a separate input. Overall, we thus have a total of at least four audio inputs, merged into an audio mixing console, which is controlled by an audio engineer eventually supported by one or more audio assistants. Its sum is fed into the audio input of the video recording console. The audio engineer also gets his or her orders from the director. Depending on the number of people in the audience and on the size of the hall, additional microphones for questions and/or for recording the atmosphere can be used.

Furthermore, for a professional production it is compulsory to provide sufficient illumination on the set. Responsible for all tasks concerning the lighting is the so-called “lighting cameraman”, who is the boss of all cameramen and all lighting assistants. They have to take care of any lighting problems, e.g., hard shadows on faces or back-light situations. Depending on the number of areas to illuminate, the number of lighting assistants for a production will change. For lecture recording, there is the need of one lighting assistant for the lecturer and another one for questioners.

Additionally, the recording of the final AV signals are monitored by an operator for the AV recording console.

Summarizing, it is obvious that a large number of pieces of equipment and cabling, together with at least three cameramen, two lighting assistants, one audio engineer, one audio assistant, one AV recording console operator and one director are a huge effort to record a lecture, apart from constraints such as the type of location and its space.

2.1.3. Important Constraints

There are many different constraints for lecture recording. Apart from reserving enough space for the crew sketched in the previous subsection, we need a separate room for the director in order to avoid disturbing the lecture by speaking into the intercom. At our university in Mannheim there is typically a 15 minute break between two lectures in the same hall; thus the equipment has to be either already installed in the hall or it must be so easy to install that these 15 minutes are sufficient for setting it up which is rather unlikely. It is obvious that just for these two first reasons, time and space, the recording effort has to be reduced. The budget is another important constraint.

Based on the requirements of the lecture hall used for our experiments, we will now assemble the camera team and the equipment that we deem absolutely necessary. For the video recording we need

- one camera set for the slides,
- one camera set for the lecturer,
- one camera set for the audience, in particular the questioners,
- one camera set for the overview,

- one cameraman for each camera set,
- one director for the entire crew,
- one video mixing console,
- one control video monitor for each camera input,
- one control video monitor for the final signal,
- one audio and video recording console.

For the audio recording we need

- one microphone for the lecturer,
- one Direct Input (DI) box for the lecturer computer's audio,
- one microphone for the atmosphere,
- one microphone per approximately nine potential questioners of the audience, mounted in a fixed position,
- one audio mixing console,
- one audio engineer.

All this is completed by an intercom system and a lot of signal and power cabling. Lighting equipment and staff can be neglected as standard lecture halls normally are sufficiently illuminated.

Based on the invoice shown in Figure 1 we calculate a total net rent of 3,830.00 € per day. The details can be found in the calculation spreadsheet in Table 1.

Table 1: Calculation spreadsheet

Position	Quantity	Description	Unit Price	Price per Position
1	4	Camera set including cameraman per day	460.00 €	1,840.00 €
2	1	Video mixing console, monitor bridge and control monitor including director per day	450.00 €	450.00 €
3	1	Audio mixing console including audio engineer per day	330.00 €	330.00 €
4	1	AV recording console including operator per day	350.00 €	350.00 €
5	4	Microphones (lecturer, slides, atmosphere, and audience) per day	35.00 €	140.00 €

6	1	AV Multi-core cable 50m per day	80.00 €	80.00 €
7	6	Intercom (radio-based) per day	40.00 €	240.00 €
8	1	Insurance for equipment per day	400.00 €	400.00 €
Total Net price				3,830.00 €

These high costs are only feasible if the lecture hall is used for recordings the whole day, not only for one lecture. As lecture days are spread over the term, it would be better to equip the lecture hall with a fixed installation and only to rent the crew. While the investment into the equipment is more than 20,000.00 €, as said in Chapter 1.3.3, the rental of the crew without any equipment would cost about 1,710.00 € per recording day, so this solution is not acceptable either in most settings.

Therefore, we propose an *automated, distributed system for lecture recording* in our approach. It is based on the human role model of the above camera team and mimics its behavior. In the next sub-chapter, we will derive the necessary parts and their functionality.

2.2. Determining the Parts of our System

Our Automatic Lecture Recording software is implemented as a distributed system in which each role is realized in a module. We will now check how the different roles can be realized in the modules.

Most obvious is the need of a *director module* as well as a *cameraman module*. The first difference is buried in the details of the cameraman: While a human camera operator is able to recognize and to interpret an action even if it is not shown in his or her camera, the virtual cameraman module is limited to events shown by the camera's view, and to status messages of the camera itself. Therefore, we add different tools integrated into one interface, the "sensor tools module". Also, a virtual pendant of the additional equipment is necessary, e.g., of the video mixing console. In the next sub-chapters, we take a more detailed look at the jobs of each member of the camera team, and show how we can realize their virtual equivalent.

2.2.1. The Director

A camera team's director of a camera team decides which scene goes "on the air". Based on the screenplay planned on events occurring during the recording and on feedback of the cameramen the director quickly (re-)decides which scene will be presented next. The director does not come to a decision at random but by taking cinematographic rules into account. These rules are binding for the director and the cameramen and they have been trained over years to be able to act accordingly. Aspects covered by these rules are, amongst others, the duration of a shot, the sequence of shots to focus on a detail, the sequence of shots to include a detail in the entire plot, preserving the line of action, the positioning of the cameras to be prepared for a shot – counter-shot scenario and the decision of how to frame a scene in order to make important parts visible.

For lecture recordings, a simple screenplay could for instance be to show the slides, the lecturer, the audience, and alternately the overview. If this sequence is done over and over again, it becomes boring, predictable, and tempts the video's spectator to expect a certain shot instead of keeping him or her focused on the contents. The director's job is to show the main action and to avoid showing the same shot or sequence of shots too frequently. As the cinematographic rules ask for certain sequences of shots, e.g., two detailed shots should be followed by a neutral shot in order to establish the relationship between the details and the whole plot, the obligation of a director is to avoid identical recurring sequences while still observing the rules.

Apart from the planned activity based on the storyboard, the director asks the cameramen for certain shots to emphasize a detail for the spectator. The other way round, the cameramen give immediate feedback to the director whether a certain shot is possible or not, either due to their position or due to the status of their camera. Typical feedbacks are messages like: "Can't see target, another object obstructs a clear view", or, "Camera is still in adjustment progress, clear view will follow". Other good reasons for deviating from the planned setting are unforeseen events and activities, for example, a question coming up. Choosing the shots is the first dimension of a director's personal style.

Another aspect is the duration of each shot. It depends on the type of production. While music clips often use a very short duration as a stylistic feature, the recom-

mended minimal duration of a shot for news production is six seconds. Documentaries or live productions often use shots with durations of significantly more than 10 seconds. Generally speaking, the duration of a shot must be at least as long as the spectator needs to perceive all the relevant details and it must end before it gets boring. This fuzziness gives room for the director's style, and it is the second dimension of creativity.

In an abstract way, a director's job can be seen as choosing the right shot out of a group of possible ones, based on cinematographic rules and in reaction to events in the environment.

2.2.2. The Cameraman

The job of a cameraman starts long before the production with a meeting of the director with all cameramen. All details of the storyboard are discussed to give the cameramen an overview over the planned production. Especially, all positions of the cameras and their moves during the time flow of the production are discussed. Some crucial shots like the starting shot and the last shot are discussed in detail, with their planned duration and whom to frame.

Right before the show, the cameramen check their cameras and give a short feedback when ready. This check includes the functionality of the aperture, the zoom, the focus, the movability, the filter, shutter settings, and other things. Then the correct position and frame for the first shot is set. When the director gives the signal, the production commences.

During the production, the cameramen have a recurring procedure to perform:

- Check the storyboard amended with the details from the meeting, for the basic points for the next shot.
- Move to your assigned position.
- Listen to the director to get the detailed orders for the next shot.
- Produce a good image of the target.
- Wait for the information that your camera is "on air".
- Hold the position.
- Wait for the information that your camera is "off air".

Depending on the space between the shots and on the prevision of the cameramen, it is not necessary that all steps are performed every time. Some of the steps may even be done in a slightly different order or even simultaneously. If any of these steps is not possible or an error occurs, the cameraman gives an immediate feedback to the director, enabling him to reconsider which camera should go on air next; this gives the cameraman time to solve the problem.

In this list, a complex task is hidden which could be described as “produce a good image of the target.” This task basically consists of the following three steps:

- Aim at the targeted part by panning, tilting and zooming the camera accordingly,
- Draw the focus on the target,
- Adjust the aperture and the shutter.

The first two steps contain the aesthetic work of a cameraman, which is also based on cinematographic rules. At the very beginning, there is the correct framing of the target, e.g., let a person look into the image; give this person enough space to the edges of the image. In cooperation with the director, it is thus possible for two cameramen to produce the images necessary for a shot – counter-shot situation, for example when two people are in a dialog. Second, the focus is drawn on the target. The chosen depth of sharpness defines to which extent the image appears to be three-dimensional. Typically the target will be shown sharply focused while the background is blurred.

Examining the tasks of the cameramen, we argue that they can be abstracted to a job consisting of three parts: a technical, an aesthetic, and a communicative one. All these parts are orthogonal to the phases “before the show”, “during the show”, and “during the shot”. Furthermore, we argue that a cameraman’s job can be described as a control-loop – work-flow as shown in Figure 6.

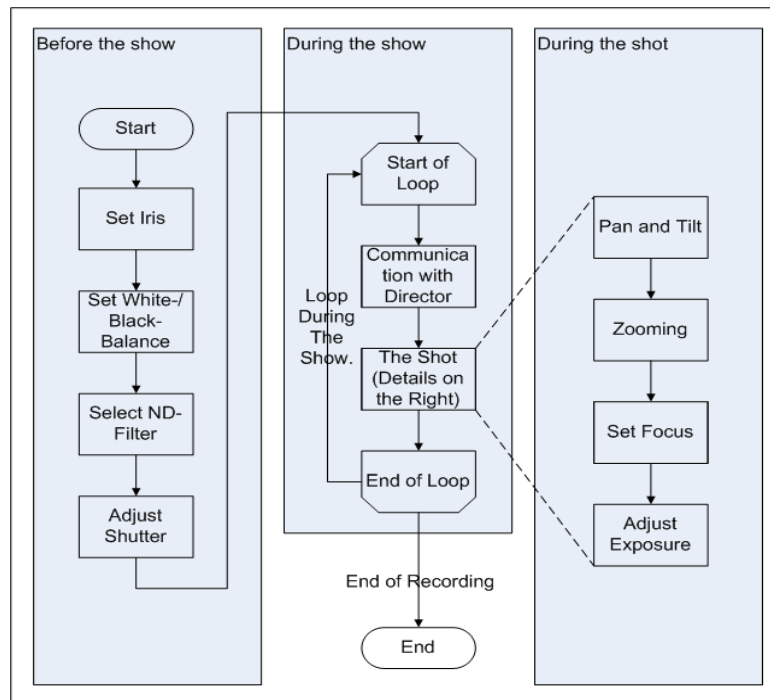


Figure 6: The job of a cameraman as a work-flow

2.2.3. Sensor Tools

We first motivate why we need sensor tools in a virtual camera team. The main difference between a human camera operator and a virtual one is the ability to decide *autonomously* how to react to the environment. A human camera operator is able to decide how to frame a certain shot, he or she can decide whether another point of view or even another shot will fit a context better. Naturally, the camera operator will arrange these things with the director but is still able to present his or her own proposals.

One simple example clarifies this: In a discussion scenario of two people, all camera operators know exactly what to do, based on the setup and on the applicable cinematographic rules. In detail, the two protagonists on the stage sit in a half-profile position so that they can see each other easily and in a way so that they can also be watched from the position of the audience. In this example, we will use three cameras: one centered camera for the long shot view, showing both protagonists sitting next to each other, one camera for the left person positioned to the right of the long shot camera, and one camera for the right person positioned to the left of the long shot camera. While this left camera will show the right person “looking from the right edge of the frame into the image”, the right camera will show the left person “looking from the

left edge into the image”. Now the director can switch among all of these shots easily without crossing the line of action between the two protagonists. Thereby, the spectator on the TV set always knows exactly where a certain person is sitting and to whom this person is speaking, even if only one person is shown in the actual shot.

The prerequisites to realize such a scenario in the virtual world are not only the correct positions of the cameras but also the knowledge which person to frame, and the knowledge how to frame this person. A virtual camera operator knows only the position of his own camera and its viewing direction. To overcome this lack of knowledge, the virtual camera operator has to be told the rest of the facts. These facts have to be collected by the virtual director and individually transmitted to the virtual camera operators. The virtual director must also be able to ask each camera operator about his knowledge but he cannot get the information of where the protagonists are sitting. This is the reason why additional sensor tools are necessary to provide the director with the required information.

In the case of a lecture recording scenario instead of a dialog scenario, the virtual director additionally needs to know a) when a questioner wants to ask a question and b) where this questioner is located. From there it is possible to set up the scenario as described above.

The task of the sensor tools module is to provide additional knowledge to the virtual director. We have already defined the necessary additional items of information for lecture recordings. That means that we have to implement at least two algorithms to get that information. As we want to minimize our effort, we define a general interface between the director and the sensor tools module, but the sensor tools module gets the data from different sensors. According to the needed information, we use an indoor positioning system as one sensor system and a question manager software suite as the other sensor system.

Indoor Positioning System

Our first sensor system has to keep up with the positions of all persons involved in the lecture. Additionally, it has to provide the virtual director with the position of the person who could be of interest as the next dialog partner.

Our favorite candidate for this sensor system was developed at our institute by Thomas King and adopted to our needs by Hendrik Lemelson (Lemelson, King & Effelsberg, 2008). It is an indoor positioning system based on 802.11 wireless LAN (WLAN). It takes advantage of the already installed access points. It enables all the devices using the WLAN in a pre-calibrated room to estimate their positions. Therefore, we have to equip all persons of whom we want to know their position with a WLAN device. In our case, we take advantage of the PDAs we already use during our lectures for interactive quizzes (Scheele *et al.*, 2003 & 2004) and which are already equipped with WLAN. Every student in our lecture hall gets a PDA handed out for the duration of the lecture. It is self-evident that the students may use their own equipment such as notebook PCs, if they have installed the necessary software beforehand. In addition, the lecturer can also be equipped with a PDA, for example the Q-Belt Integrated Computer (QBIC) wearable computing device see (Büren von, 2002) and (Amft *et al.*, 2004) developed at the ETH Zürich, in order to track him or her if he or she moves around. Now, we are able to get the positions of the involved persons in the lecture at any time.

Questions and Answers

One point in time when we certainly need to know a person's position is the moment a questioner wants to ask a question as we want to make a virtual camera operator aim at him or her.

Besides the coordinates, the director needs the information that a questioner wants to ask a question. In addition, he needs to be informed when the lecturer starts answering the question, whether the questioner asks additional questions, and when the question is finally answered; this information is needed in order to switch from the standard lecture context to the dialog context, and to inform the camera operators accordingly. As we need this information as precisely as possible, a question manager (QM) software suite is used to map the question – answer work-flow. This software suite is based on the Client-Server paradigm, in which we use software on the PDAs handed out to the students as clients and software on the virtual director's machine as a server. In this case, another kind of client is the software on the lecturer's computer in order to let him or her control the question – answer interaction.

In a standard lecture, questions are important in many ways. They show that students pay attention to the lecture. Moreover, they often encourage others to think about the questions as well as about the topic itself. Finally, they give the lecturer feedback about the questioner's, and sometimes even of the audience's, level of understanding.

Let us now have a closer look at the question – answer interaction itself. What are the different phases of this interaction? What types of interaction do we have to expect? Which tasks must be transferred to our system and which may be neglected? In the first phase of the work-flow of a typical question and answer session during a lecture, a questioner raises his or her hand, tries to draw attention, and waits for being given the floor by the lecturer. In the next phase, the question is asked and then the lecturer answers. In many cases, the interaction is over with the lecturer having answered. So, the basic question – answer interaction consists of the following five steps:

1. The questioner announces a question by raising his or her hand.
2. The lecturer gives the floor to the questioner.
3. The questioner asks the question.
4. The lecturer answers the question.
5. The question – answer interaction is finished after the question has been answered.

Figure 7 shows these five steps as an interaction diagram between lecturer and questioner.

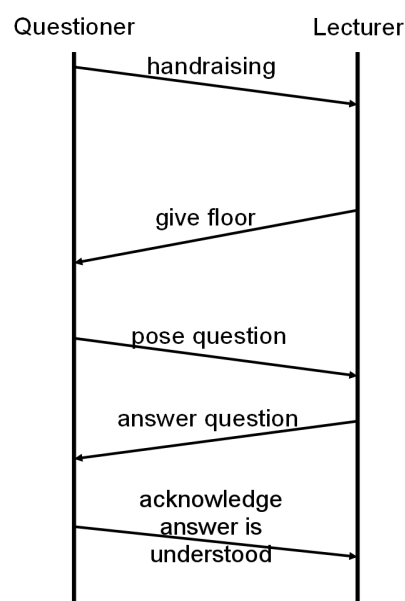


Figure 7: Interaction Diagram of Lecturer and Questioner

Nevertheless, sometimes the questioner does not understand the first answer, or additional questions about other details arise so that the primary answer does not meet the questioner's expectations. Under such circumstances the questioner insists on further clarification. Instead of acknowledging the answer, a further inquiry arises which again needs to be answered.

Even if it seems that question and answer sessions can be easily described by this interaction diagram, there are several possibilities of how such a scenario might evolve. One might for example think of a discussion starting when a new participant, maybe a fellow student, joins the scenario, and a discussion starts. In the best case, it is a highly fruitful discussion. On the other hand, this can also lead to a controversy which the lecturer has to stop. In between, there are many different possibilities involving a number of different people or leading to any direction or detail of topics. A lecturer may not only in the worst scenario have a good reason to stop the interaction, e.g., for didactic reasons to keep a planned order of details, for giving the floor to another questioner, or for starting another explanation of the relevant part. Furthermore, the lecturer must be able to defer a question for a certain time or to even deny asking questions at all, depending on the progress of the lecture. It is thus obvious that not every question – answer interaction will work in paired phases.

In order to counteract an unmanageable variety of different interaction scenarios, we have analyzed the different tasks of interactions. By determining recurring tasks, we are able to reduce the complexity by mapping the different interactions into well-arranged pieces of software. Every question – answer interaction can be divided into two parts. The first part is to control who is allowed to speak; the second part is either “speaking” or “listening”. Table 2 shows the aggregated actions assigned to the first part:

Table 2: Aggregated Actions Controlling the Interaction.

- Perceiving (re-)announcement.
- Giving the floor.
- Giving the floor to the lecturer.
- Deferring giving the floor for a certain time.
- Denying giving the floor.
- Ending an interaction / a discussion.

The main aggregation is to abstract the addressee of the action; It does not matter whether the original questioner or a participant joining later is the addressee of the action. This is true for all actions besides “giving the floor to the lecturer” and “ending an interaction / a discussion”. In order to control even complex interaction scenarios, it is necessary to keep track of the participants and the actions addressed to them.

If we neglect the possibility of participants joining lately, the number of possible tasks stays the same but the effort of keeping track of the number of addressees is reduced significantly. We therefore reduce the number of possible addressees to one questioner for our prototype.

As the whole interaction is mostly performed by speaking, it is really important that the audio is clearly audible. In contrast to the common lecture habit, not only the voice of the lecturer but also the voice of the questioner needs to be understandable in order to enable others to keep up with the context. While a lecturer is used to speak loud and clear a questioner may need to be encouraged to do so. We will now present the role of the sound engineer who controls each single sound source to be easily understandable.

2.2.4. The Sound Engineer

The sound engineer of a camera team is responsible for the correct deployment and use of the microphones and, during the production, for the correct levels and the sound and mix of the audio signals. Before the production, he or she spreads out the microphones and fixes them at the foreseen positions. These positions can be microphone stands, “boom poles”, or even a person in case of wireless clip-on microphones, for example.

While the lecturer does normally get a headset or clip-on microphone, the audience needs a different approach. For the atmosphere sound, omni-directional microphones are used, but for questions it is necessary to use cardioid microphones put closer to the questioner. As it is very seldom to have a room equipped with ceiling-mounted microphones for four to nine people, two solutions are common: one is to hand out one or more wireless hand microphones, which always takes a lot of time and sets up a high barrier questioners need to overcome; The other solution is to set up one microphone for every participant who is a potential questioner. This needs more initial ef-

fort but makes it possible to record questions without any delay and is a non-intrusive way for the questioner.

During the production, the sound engineer controls the volume levels of all signals, mixes the signals into logical units, e.g. an audience unit, an atmosphere unit, a lecturer unit, and a computer sounds unit. By checking the units' sound and volume levels again, he or she mixes the units down to a final stereo mix, for example. Depending on the size and complexity of the production, a sound engineer employs assistants and realizes the work-flow in a hierarchical way.

For our approach, we exploit the fact that we have already handed out PDAs to the students and that each PDA has a built-in microphone. Therefore, we are able to easily use a non-intrusive way of handing out microphones to all our potential questioners. We implement a piece of software to record the audio on the PDAs and transport the sound from them into the central unit, joining the audience-based sounds into one single audience audio stream. This is done by the so-called virtual audio engineer. In the case of more than one single questioner, the audio engineer has to mix all sounds, in case of many possible audio sources he only has to switch the channel according to the active questioner. But in any case, he has to control the resulting volume level.

We use a hierarchical approach for our prototype, i.e., there is a virtual sound engineer responsible for the correct sound of the audience and a master engineer responsible for the final mixing of the lecturers' audios, the audiences' audios, and computer-generated sounds. In order to keep the effort for our prototype system small, we have waived implementing separate audio assistants for the lecturer's voice and the computer sounds as the lecturer is used to speak loudly and clearly, and as it is easy to tune the volume level of a computer accordingly.

As the sound engineer of the audience depends directly on the active state of the QM server software and the master engineer is responsible for the final audio output, we have implemented these routines directly in the QM server software and in the audio/video (AV) mixer software. Nevertheless, it is possible to separate these tasks into specialized audio engineer software if necessary.

2.2.5. The Lighting Technician

The task of lighting technicians is to provide the best possible lighting. There are different types of light and different tasks to master: One has to differentiate between ambient light and directed light, between natural and artificial light, and between useful and troublesome light.

Natural sunlight can be seen as the most beautiful type of light, but its disadvantage is that there is no guarantee that it is available at a certain moment, in a certain intensity, etc. Therefore, we need to amend a scene with artificial light in order to be independent of a natural light source.

For basic illumination, we need a homogeneous ambient lighting which is individually enhanced by directed spots for the more prominent places on the scene. Typically, the protagonists are put into this light to make them appear more prominent; this also compensates any troublesome lighting conditions such as back-light, for example. Depending on the number of protagonists and the naturally changing lighting conditions the task of the lighting technician can be very complex. But fortunately, as we are located in a lecture hall, the basic illumination is already very good, and a back-light compensation can be provided by the camera operators changing the iris setting of the cameras. Therefore, we have decided to not implement a lighting technician for our prototype.

2.2.6. The Audio/Video Mixing Console

The work-flow chain of our distributed automatic lecture recording system prototype is almost complete. All necessary team members are defined. The last missing part is an instance taking the orders from the director to select, switch, fade, or mix the desired video signal, to mix the incoming audio signals, and to output the resulting AV signal for recording or streaming. In the real world, this instance is not a person but an AV mixing console, operated by a person.

As our prototype is a piece of software, we need to build an automated AV mixing console in software, controlled by the director. In our scenario, the AV mixer needs to have four video stream inputs using the *MPEG-4 part 2 codec* and up to three audio stream inputs using the *μ -law codec*, all of them via *RTSP/RTCP/RTP streams*. These technical requirements come from the available AV servers. Every audio and video

signal is converted through these servers into streams, transported over IP through the LAN. The AV mixer needs to have another IP input for the director's command messages. Its output is either a file containing the resulting AV data, or an AV stream for live distribution.

As all the data for input and output are transferred via LAN, the AV mixing console computer can be located in any place on the Internet. Thus, it is easy to uncouple and distribute the workload from the place where the data originates.

The recorded lectures are afterwards prepared for their use as a video-on-demand service in many different formats to be able to support different end user devices. We do not primarily aim at live streaming but at recording the lectures.

2.3. System Overview

After having determined the parts of our prototype of the distributed Automatic Lecture Recording system, we now give an overview of the components' interrelationship. We start with a diagram of the different components and the communication channels between them, as shown in Figure 8.

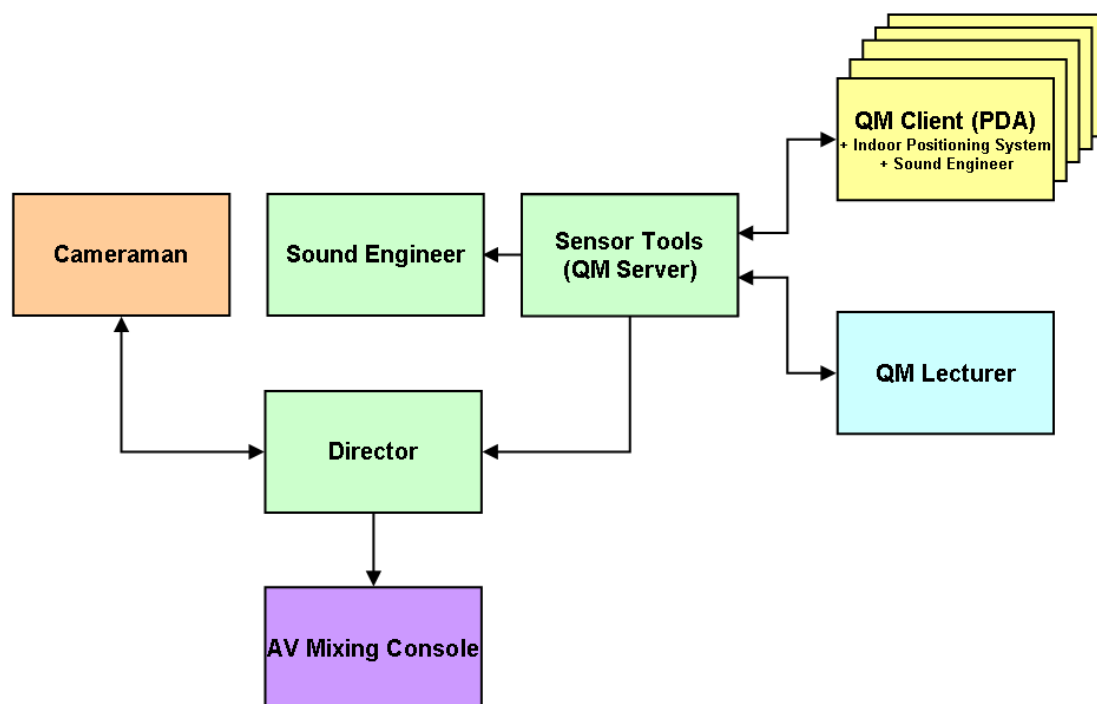


Figure 8: Systems and Communications Channels Overview

The different colors in the figure symbolize the different computers on which the components run in our prototype environment. This is only an example as all commu-

nication channels use Text/XML messages over IP technology, whether they are on the same computer or not. So, depending e.g., on the performance capabilities of the machines used, one can combine or separate the components on several machines.

In order to give an understanding for the whole prototype testing scenario and its complexity, we show a draft of the prototype system with all instances and any used data connections in Figure 9.

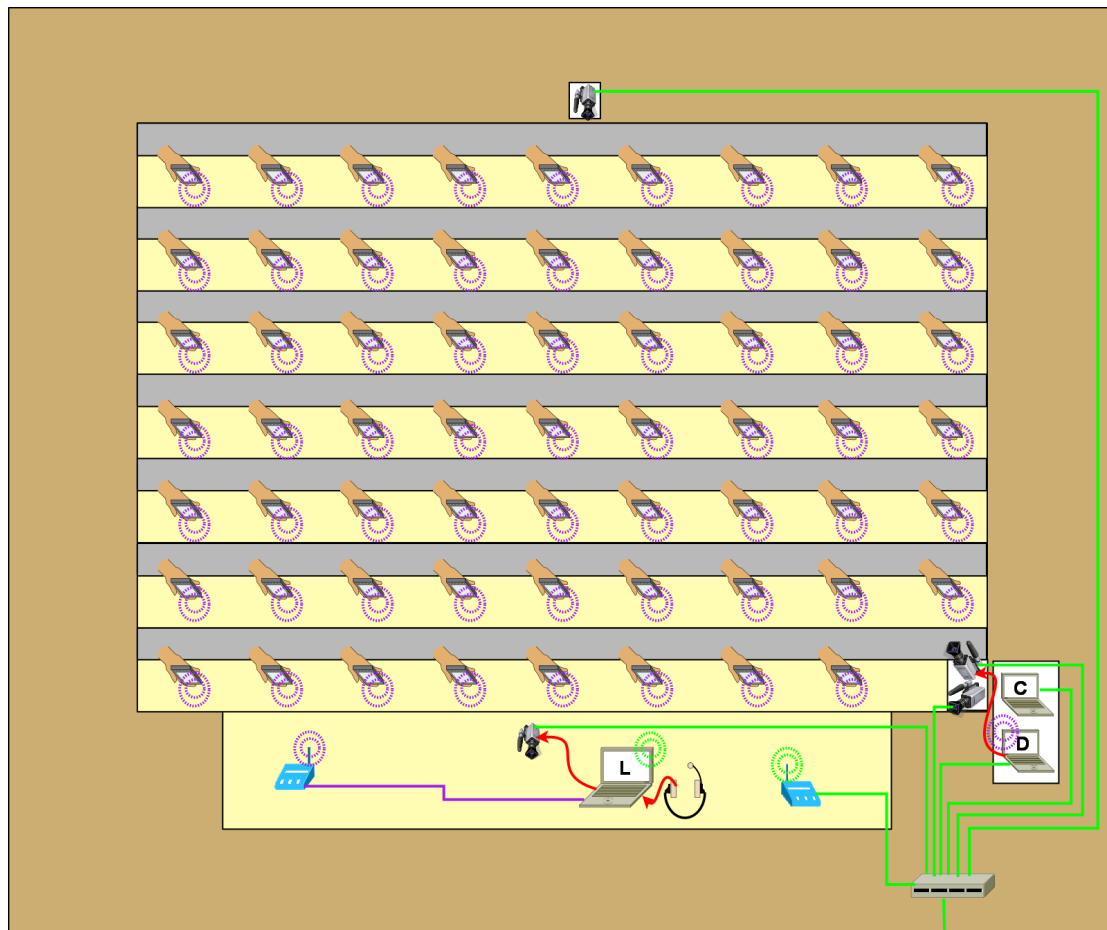











Figure 9: Draft of the Prototype System

Legend:

	Camera/Video-Server including MPEG-4 and μ -law encoder. RTP over IP via RJ45 port.
	Wireless LAN (802.11) Access Point.
	Notebooks: L stands for Lecturer , D stands for Director , and C stands for Cameraman .
	PDA used by a student.

	Network switch providing the connection to the university's network.
	Headset of the lecturer.
	Wired or wireless LAN connection in the IP range of the university.
	Wired or wireless LAN connection in the IP range of the question and answer (QA) management software suite.
	Analogue audio connection.

2.3.1. Virtual Director's Main Ideas

Derived from real recordings, the *director* plays the central role, communicates with the camera team, and decides which camera goes on air. The director is based on an extended FSM. Its transitions are not based on fixed probabilities but on conditions generating numeric values at runtime which lead to changing probabilities for each transition. In contrast to fixed values, this procedure guarantees specific probabilities depending on the current conditions of the sensor tools. The FSM has three contexts: a lecture context, a question context, and an answer context. This gives us the advantage that states showing the same shot but being located in a different context can be reached by different transitions, and therefore the conditions defined for those transitions may differ. When the duration of the previous shot ends, the transition to the next shot will be selected based on the current probabilities.

In detail, our approach starts by determining all transitions going out from the currently active state and initializing their probabilities. During the next steps, these probabilities are modified: They are decreased, first, depending on how recently the possible new state was active. Then, the sensor inputs are evaluated; they influence the transition probabilities, e.g., if a sensor detects that someone wants to ask a question. Also, we take the visual activity in each camera output into account and modify the belonging probabilities accordingly. Finally, the highest value of all probabilities is determined, which corresponds to the transition providing the best reaction to the environmental inputs taking the built-in cinematographic rules into account.

Additionally, the duration of the next shot is determined based on a time interval, individually assigned to each state. The basic rule is that shots of complex scenes get a longer duration than scenes with less complexity. The resulting duration of a shot lies

within this time range. If necessary, time can be added to the duration depending on the motion intensity of the scene.

Overall, this procedure provides a very fine granularity to describe the conditions to reach a certain state in a certain context. Its granularity and its responsiveness to the environment using the mentioned probabilities lead to decisions which are often similar but seldom identical in similar situations. As the description of the FSM is hard coded in an editable XML-file, it is easy to adopt it to different tasks. Therefore, the know-how of basic cinematographic rules is necessary in order to define the probabilities successfully. Figure 10 shows a graph of an exemplary FSM for standard lectures:

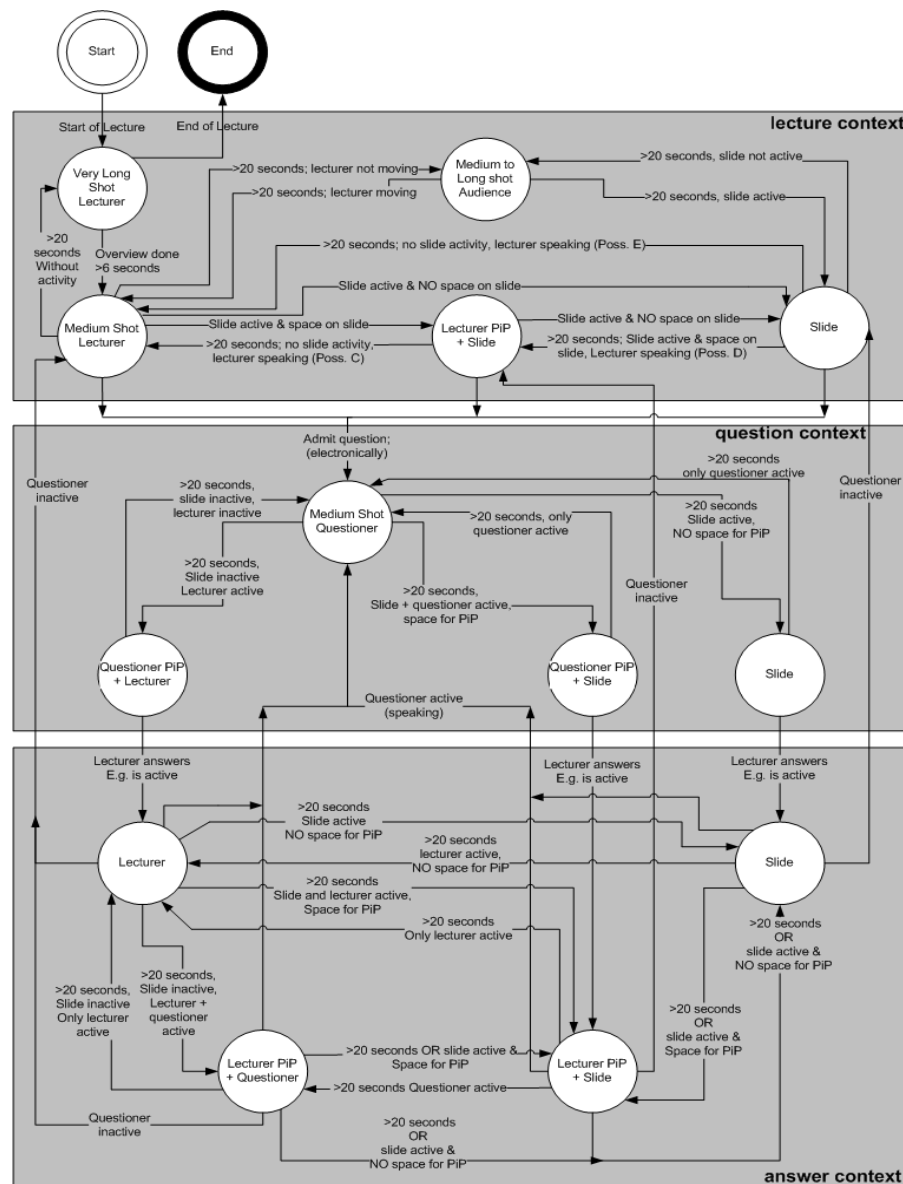


Figure 10: Graph of an Exemplary FSM for Standard Lectures

2.3.2. Virtual Cameraman's Main Ideas

The virtual *cameraman* is based on its real world role model. As we said earlier, analyzing the job of a cameraman leads to three main tasks: a technical task, an aesthetic task, and a communicative task. In Figure 6 we have described his or her complete job as a control-loop in order to provide a work-flow for its implementation.

While it is obvious for the technical task how to be implemented, it is not easy for the communicative task and definitely not clear for the aesthetic task. In order to make computers playing as a team, we have decided to distribute the responsibilities: Each virtual cameraman reports the state of the camera to the director. He also reports the motion rate of the video shown, which enables the director to get a more global knowledge. Using it, the director gives detailed orders to the respective cameraman. Typical examples are that he decides to frame a person close to the left edge or the right edge of the image, that he decides whether slides have been changed and therefore the new slide should be on air next, and that he decides how to react to the information that a camera is not able to provide a demanded shot at a certain moment.

So, the communicative task has been extended for the virtual cameraman compared to the human one, as it is necessary to provide status reports actively. Nevertheless, the basic communication is still important and contains messages like the respective on-air states, like “the camera is still in motion”, like “zooming in progress”, like “drawing the focus in progress”, like “exposure adjustment in progress”, etc.

2.3.3. Sensor Tools' Main Ideas

As a virtual cameraman is not able to reliably detect a questioner raising his or her hand, and as he is not able to reliably determine his or her position by itself, a complementary tool set is needed for our distributed system. In order to provide a base for different sensors developing over time, we have decided to set up a framework for the different types of sensors. This framework only sends three values to the director: an event flag which marks a sensor input as requiring an immediate reaction, a composed string value describing the sensor input and its parameters like intensity or position, and finally a description of the camera which is intended to be on air next, i.e., *Lecturer*, *Slides*, *Audience*, or *LongShot*.

The basic sensors implemented in our prototype are a question – answer (QA) management tool, mapping the interaction between a questioner and the lecturer to the appropriate sensor values, and an indoor positioning system determining the position of a questioner in the lecture hall. As both tools need to collaborate, we trigger the indoor positioning system by the QA management tool when a questioner announces his or her question. All the resulting values are prepared to fit into our framework. Naturally, the QA management software extracts all the information needed out of these values, too.

The QA management software was newly built for our distributed system, while its hardware was adopted from the WIL/MA project from our research group (Scheele *et al.*, 2004). The Indoor Positioning System was also developed at our research group by Thomas King, it has been adapted to our needs by Hendrik Lemelson.

2.3.4. Virtual Sound Engineer’s Main Ideas

The sound engineer of our distributed system should be able to include all standard sound sources of a lecture. This includes the commonly recorded voice of the lecturer, but also computer-based sounds from the lecturer’s computer and the voices of questioners in the lecture hall. We wanted to overcome the gap of the changing lecture context arising from a question which was not recorded properly up to now: the spectators of the resulting video were not able to hear the question and therefore were often forced to guess it out of the given answer unless it was repeated by the lecturer.

In many approaches, different setups of spread-out microphones have been tested and can be classified into three cases: a) an individual microphone for each potential questioner, b) some omni-directional microphones spread out over the lecture hall to record all hearable sounds, and c) some dedicated microphones on stands which a questioner has to approach to ask his or her question.

While solution C induces a very high psychological barrier to ask a question, it is cheap to realize; solution B is also cheap and does not set up a barrier but suffers from ambient noises so that the question is seldom clearly audible. At last, solution A provides a clear sound, it is a non-intrusive solution which leads to no psychological barrier; however it needs a huge effort concerning the equipment.

By using the built-in microphones of the PDAs already used in the WIL/MA project, we are able to reduce the cost of solution A significantly, still providing each potential questioner with his or her individual microphone. The sound can be recorded in a very good quality and transported via wireless LAN without any noticeable loss in quality.

2.3.5. AV Mixing Console's Main Ideas

As all video and audio sources are transported over IP using the RTP protocol, it is theoretically easy to synchronize them. However, the cameras and streaming servers we have used to stay inside our budget plans unfortunately do only provide “up to 30 frames per second” (fps). This does not mean that such a streaming server is able to provide 25 fps easily and constantly, but that the variable frame rate can reach up to 30 fps. In reality, we have measured about seven fps for the long shot camera and its streaming server while the lecturer and the slides provided about 20 to 22 fps.

We have therefore implemented a buffer for each single video stream which is always filled with a valid image. Now we are able to synchronize the four incoming video streams and to generate precisely 25 frames per second for the resulting video, at the risk of doubling single frames. Having laid these foundations, we have implemented the typical functionality of a video mixing console of switching, fading, and blending videos, including the special effect of producing a Picture-in-Picture (PiP) video in real-time.

Similar to the streaming server's constraints of video, there are constraints concerning audio. As mentioned, our audio sources can only provide an “a-law” or “μ-law” coded audio stream with an 8 bit quantization and an 8000 Hz sampling rate. These audio stream characteristics might be sufficient for the human voice. They originally stem from the transmission of voice over telephone lines but do not provide a clear and brilliant sound as one would expect in AV recordings. As already mentioned we could not exceed our budget and therefore accepted these constraints for our prototype. However, we have decided to abstract the interfaces for the cameras and the sound engineer so that we can exchange the equipment easily with a minimal coding effort.

The lecturer's sound characteristics and the computer-based sounds are typically at 16 bit, 44.100 Hz in stereo before fed to the streaming servers. The questioners' sound characteristics are at 16 bit, 22050 Hz in mono before fed to the streaming server. At

the AV mixing console all sounds are decoded to 16 bit, 8000 Hz in mono, mixed together and up-sampled to 16 bit, 48.000 Hz in stereo for the resulting audio stream.

Finally, the resulting video and audio streams are transformed into a Digital Video (DV) in an Audio-Video-Interleaved (AVI) container format, a so called DV-AVI file on disk which can be transcoded to different streamable formats by arbitrary transcoders.

3. System Implementation

After having defined the components for our distributed Automatic Lecture Recording system prototype, we are now going into details concerning the implementation of each component. We show the tasks that each component has to fulfill as well as our approaches to achieve this. We also present code details when useful.

3.1. Director Module

The director as the central instance of our system is responsible for the selection of a shot, determination of its duration, and the transition to the next shot including all necessary details. Its decisions should not be done at random but based on cinematographic rules.

Cinematographic rules do not only apply during production but also beforehand for the correct set-up of the scene. It is necessary to choose the location of the cameras carefully. At first, they should be located in a way so that backlight situations are avoided as far as possible; at second, they should be located so that the line or lines of action that arises during the production will never get crossed in order to avoid any disturbance of the spectators. These two rules have to be fulfilled “manually” while setting up the cameras in the lecture hall as long as we do not use autonomous cameras which would be able to move on their own. Nevertheless, at least the coordinates and the standard directions have to be fixed manually and noted down in the camera-men’s configuration files. An example of such a configuration file is shown in Appendix 7.1.2.

During the production, many more cinematographic rules apply. The most basic ones determine the type of a shot, the order how different types of shots have to be chosen, and the duration of each shot. In the first moment, we think of fixed values and rules which can easily be translated into a software system. Unfortunately, the rules are in fact very fuzzy. In the next paragraphs, we will show the range out of which to choose and how to choose the appropriate parameters.

The different types of shots vary according to (Thompson, 1998) from a very long shot giving an overview of the whole scene to a very close shot showing a small detail, e.g., of the lecturers’ face. It is obvious that a recording starting with the close-up of the lecturers’ mouth would be very surprising as no one is able to know whose mouth it is and in which context it is recorded. Therefore, in most cases, a production

will show a very long shot in the beginning to give an overview of the scene and to introduce the spectator into the context. From there on, certain parts of the whole scene can be shown step-by-step, going more and more into detail. There is an interesting exception: When showing a detail of one person, it would be quite disturbing to show the same or a similar detail of another person right afterwards. A so-called “neutral shot” has to be shown in-between to make the change of the protagonist clear to the spectator. The neutral shot in this case is any shot showing no person at all or a whole group of people as a very long shot. Generally spoken, there is a rule saying that a neutral shot should be inserted about every two or three shots, a rule which again is not carved in stone.

A special version of switching back and forth between two shots showing similar details is the so-called “shot – counter-shot” scenario, e.g., used for switching between two discussing persons viewing each other. In this case, it is necessary to directly switch between the two cameras framing these two persons while one is framed so that he or she is placed on the left edge looking into the right half of the image and the other person is framed so that he or she is placed on the right edge looking into the left half of the image. The resulting view gives the impression that these two persons face each other even though there might be thousands of kilometers between them. Nevertheless, from time to time, the neutral shot has to be shown as mentioned before. If both people are really sitting in the same room, the neutral shot will typically show both of them in the scene.

After the director has chosen a shot, the next question is how long to show it. As a matter of course, there are fuzzy cinematographic rules for this. At first, there are different genres which lead to different time ranges of shots. Very short durations are often used as a stylistic device in music videos while very long durations are often established in laudations or other speeches. A typical time range for the news genre is that a shot should be shown for about six seconds, with an absolute minimum of four seconds and a maximum of about ten seconds. Overall, not the number of seconds is important but the time a spectator is given to perceive the shot. So, the rules say that the minimal duration of a shot must give the spectator the chance to perceive all details of the shot, and the maximal duration is reached when the shot gets boring.

As lectures are a special genre similar to speeches on the one hand but also similar to news on the other hand, the time ranges for a shot are set from six to 90 seconds. In

order to avoid boring shots, we assign a recommended duration to each shot. These durations have to be set manually according to the genre, the shots, and their contexts. Another factor limits the duration of a shot: in case of an event happening which is important to the production, the camera showing it has to be switched on air as soon as possible. In any case, such an event has to be taken into account when choosing the next shot.

This is necessary in case of an event, but also in time before the duration of the earlier shot has expired. It can be a shot of the same camera using another zoom or direction in case of the audience camera. However, in most cases, it should be a new point of view generating a completely new visual stimulus, e.g., showing the slides after showing the lecturer.

Here, another cinematographic rule applies: one should not use the same sequence of shots over and over again. If the same sequence of shots is used several times, the spectator will detect this rhythm and consequently will not focus on the content of the shot but whether they are right in predicting the next shot. While for normal TV productions this indicates simply a bad director, for lecture recordings it is counter-productive as it detracts the viewer's attention from the content.

Of course, the number of cameras is limited, and therefore the number of different shots is limited too, which leads more or less to predictable sequences. However, we have to distinguish between two types of cuts between shots even if they are showing the same content: an "unmotivated cut" and a "motivated cut". An unmotivated cut is simply a tool to avoid one shot taking too long. A motivated cut is one, where a specific action induces the switch to the next shot, e.g., in order to show this action in more detail. As the motivated cuts follow interesting actions and therefore satisfy the spectator's curiosity, it looks to him or her as a completely new shot even if the same camera is often on air. Therefore, reacting to the environment is one of the most important cinematographic rules. Good examples for environments that a camera team usually reacts to are gesticulating lecturers, a noisy audience, a questioner posing a question, or a slide getting annotated by the lecturer. In the case that such an action happens, the director tries to show an appropriate shot. If such an action is shown immediately, it is easier to understand the entire scenario. However, there is a restriction: in case that the director has just switched to a new shot, he or she will wait a reasonable time to let the spectators perceive this shot before he or she reacts to the envi-

ronment. Normally, nothing of the planned or expected actions is so important or urgent to skip an active shot; only exceptional circumstances could be a reason to do so. These are the basic cinematographic rules we want to implement in our prototype. They considerably increase the possibilities for diversified reactions of a director, in contrast to a system neglecting such rules. Their application should be sufficient to realize a first imitation of a human director for the use in lecture recordings. It is not possible to implement a virtual director being prepared for any task, situation or location in the same way as a human director, but it can fairly well replace him or her in standard situations of frontal presentations.

3.1.1. Tasks to Fulfill

In the previous section, we described the cinematographic rules we want to implement. Technically spoken, a shot can be seen as a state in a Finite State Machine (FSM). To be more precise, we use an Extended Finite State Machine (EFSM) as we amend properties of each state and the transitions between the states do neither use fixed values nor binary decisions to trigger them. While in an FSM the transitions are associated with sets of Boolean input conditions and sets of Boolean output functions, the transitions in an EFSM are expressed by sets of trigger conditions, used like if-statements and variables. A transition fires if all according trigger conditions are satisfied.

Generally an FSM is defined by a quintuple $A = (S, I, F, O, \delta)$ where:

- S is a finite, non-empty set of states,
- I is a finite, non-empty set of symbols as the input alphabet,
- F is the set of accepting states, a possibly empty subset of S ($F \subseteq S$),
- O is a finite set of symbols as the output alphabet,
- δ is the state-transition function: $\delta : S \times I \rightarrow F$.

Definition/Formula 1: Definition of a Finite State Machine.

Sometimes, a starting state s_0 is also given, according to (Brauer, 1984). As the behavior of an FSM is strictly dependent only on the according input and the given state transitions, FSMs tend to need a high number of states in order to do their goal as soon it gets more complex. Therefore, their specifications get extended and result in

so-called Extended Finite State Machines (EFSM), as mentioned in (Effelsberg, 1998). They get extended by adding variables and parameters, conditions, and commands. The variables and parameters hold additional information not necessarily important for the process of the automaton.

While in a pure automaton the state transitions are done only dependent on the according input symbol, inside the extended automaton the values of the variables and parameters need to be taken into account additionally being used inside conditions. The conditions are assigned to transitions. A transition fires if its condition is fulfilled. Finally, in an EFSM not only an output symbol is generated but also the values of the variables may get changed. Therefore, commands can be assigned to transitions.

According to (Leue, 2000) is an EFSM defined by this tuple: $EA = (S, D, V, O, I, T, C)$, where:

- S is a finite, non-empty set of symbolic states,
- D is a n -dimensional linear space, each D_n is a data area,
- $V = \{\Pi, v_1, \dots, v_n\}$, a finite set of program variables, where:
 - Π is the control variable on the domain S ,
 - $\{v_1, \dots, v_n\}$ are data variables,
- O is the finite set of output signal types,
- I is the finite set of input signal types,
- T is a transition relation: $T : S \times 2^D \times I \rightarrow S \times 2^D \times O$,
- C is a start condition over $S \times 2^D$.

Remark: “state” is a function: $s : V \rightarrow 2^S \times 2^D$.

Definition/Formula 2: Definition of an Extended Finite State Machine.

The cycle behavior of an EFSM consists of three steps:

1. Evaluate all trigger conditions used as inputs to the second step.
2. Compute the next state and the signals controlling the last step.
3. Perform the data operation(s) associated with the transition.

So, we amend the states with data variables, the transitions with trigger conditions and control variables, and realize the evaluation block and the arithmetic block by the program logic necessary for our system.

The properties we amend to each *state* describe the shot in more detail. They consist of the shot's name, a context to which this state belongs to, a value which camera is necessary to do this shot, an optional value whether or not a PiP-camera is necessary for this shot, and finally all transitions going out of this state.

The *transitions* are divided into two groups: On the one hand, "unmotivated" transitions when the duration of the last shot has expired, on the other hand, "motivated" transitions if events require an immediate reaction. Both types of transitions do have conditions which naturally differ, depending on the actual membership to one group. In addition, conditions leading to the same shot may differ depending on the context of the lecture. There are different conditions to switch to the audience in case of the "question context" compared to the standard "lecture context". So, we have amended the FSM with contexts in which the same cameras are used but in a different manner, so that different shots and therefore different states are necessary.

A good example is the audience camera. Normally, it shows the whole audience in the lecture hall from the front left side, but, in a question context, this camera zooms in on the questioner sitting in the audience. The shot shown has changed completely. Furthermore, increasing the number of states by introducing contexts enables us to define the conditions for a transition with a finer granularity. The result is a larger variety of possible transitions coming from a state. Enlarging the number of possibilities how to reach a state is one step to make the FSM less predictable.

We did not stop there but further increased the number of shots. Coming from the basic four shots from our four cameras, we started to combine these basic shots to a picture-in-picture (PiP) image, e.g., showing the slides with the lecturer picture-in-picture.

This is the resulting tuple of variables describing a transition:

Transition := (State_{Orig}, State_{Target}, T/E, C_{txt}, Shot, Z, H/F, Cond_{Lect},
Cond_{Slide}, Cond_{Aud})

Legend:

State_{Orig} = Originating state
 State_{Target} = Target state
 T/E = Time transition or Event transition
 C_{txt} = Context
 Shot = Shot (Lecturer, Slides, Audience, LongShot, PiP...)
 Z = Zoom-factor
 H/F = Hard Cut or Fading
 Cond_{Lect} = Conditions concerning the lecturer
 Cond_{Slide} = Conditions concerning the slides
 Cond_{Aud} = Conditions concerning the audience or the questioner

Definition/Formula 3: Definition of the Transition - Tuple.

The FSM continuously loops through the decision process in which all the different inputs are taken into account. In detail, the virtual director has to process the following tasks:

- Choosing the active shot.
- Choosing its duration.
- Processing the input values and events of the virtual cameramen and the sensor tools.
- Choosing the transition to the next shot.
- Giving the resulting orders to the cameramen, to the AV mixing console, to the sound engineer, and to the QA management software.

3.1.2. Implementation Details

Coming from this abstracted view on the virtual director's tasks, we will now go into its implementation details. We follow a typical life cycle starting with loading the FSM description into the memory, establishing the communication with all partners, entering the main loop via the starting state, and presenting the steps of the main loop: processing all input values, feedbacks and events, going into determining the next transition, generating the resulting commands for the various receivers; finally leaving the loop when receiving the "End of Lecture" signal, reaching the end state and communicating this before terminating the virtual director program.

As different lecturers have different styles of teaching the FSM should be flexible enough to be adapted to different needs. So, a hard-coded design is not useful. We therefore decided to use a configuration file written in XML. The file starts with some metadata:

```
<?xml version="1.0" encoding="UTF-8" ?>
<FSM>
  <Name>Lecture</Name>
  <Version>0.2 Draft</Version>
  <Description>Basic Lecture Recording Draft</Description>
  <Author>Fleming Lampi</Author>
  <Date>y2006.m04.d05</Date>
```

...

At next, the communication partners and their connection details are set:

...

```
<Director>
  <IP>134.155.92.68</IP>
</Director>
<Cameras>
  <Camera>
    <Name>Audience</Name>
    <RTSP>rtsp://134.155.92.47:1026/mpeg4/1/media.amp</RTSP>
  </Camera>
  <Camera>
    <Name>Lecturer</Name>
    <RTSP>rtsp://134.155.92.23:1024/mpeg4/1/media.amp</RTSP>
  </Camera>
  <Camera>
    <Name>Slides</Name>
    <RTSP>rtsp://134.155.92.80:1027/mpeg4/1/media.amp</RTSP>
  </Camera>
  <Camera>
    <Name>LongShot</Name>
    <RTSP>rtsp://134.155.92.74:1025/mpeg4/1/media.amp</RTSP>
  </Camera>
</Cameras>
<Recorder>
  <IP>134.155.92.12</IP>
  <Port>49901</Port>
</Recorder>
```

...

The director gets the information which network interface does the communication, what the names of the cameras are and their respective RTSP-Uniform Resource Locators (URL), and what the IP address and the port of the recorder are, i.e., the IP address for the AV mixing console.

Now, all prerequisites of the FSM are defined: the contexts in this extended FSM, all allowable types of cuts between two shots, all possible instances of conditions, and all objects on which these conditions may be applied. At this point, we only show some

examples per group while the whole definition file can be found in the appendix (see Section 7.1.1):

```

<Contexts>
  <Context>
    <Number>0</Number>
    <Name>out of context</Name>
  </Context>
  <Context>
    <Number>1</Number>
    <Name>lecture context</Name>
  </Context>
...
</Contexts>

<CuttingTypes>
  <Type>
    <Number>1</Number>
    <Name>Cut</Name> <!-- Schnitt -->
    <SourceChange>Yes</SourceChange>
  </Type>
  <Type>
    <Number>2</Number>
    <Name>Fade</Name> <!-- Blende -->
    <SourceChange>Yes</SourceChange>
  </Type>
...
</CuttingTypes>

<ConditionTypes>
  <CondType>
    <Number>0</Number>
    <Name>time</Name>
  </CondType>
  <CondType>
    <Number>-1</Number>
    <Name>still</Name>
  </CondType>
  <CondType>
    <Number>1</Number>
    <Name>gesticulating</Name>
  </CondType>
...
</ConditionTypes>

<ConditionObjects>
  <CondObject>
    <Number>1</Number>
    <Name>lecturer</Name>
  </CondObject>
...
</ConditionObjects>
...

```

Now we define the states and transitions of the FSM. Again, we only present snippets here, for details see Section 7.1.1.

```

...
<Definition>

```

```

<Startstate>1</Startstate>
...
<State>
  <Number>2</Number>
  <Name>Very Long Shot Lecturer</Name>
  <Context>1</Context>
  <Camera>LongShot</Camera>
  <CameraPiP>Empty</CameraPiP>
  <Transitions>
    <Time>
      <min>15</min>
      <recommend>18</recommend>
      <max>30</max>
      <randomRange>20%</randomRange>
      <Possibilities>
        <Possibility>
          <Number>1</Number>
          <Type>Time</Type>
          <NewState>3</NewState>
          <CutType>1,2</CutType>
          <Conditions>
            <lecturer>moving,active,speaking</lecturer>
          </Conditions>
        </Possibility>
        <Possibility>
          <Number>2</Number>
          <Type>Time</Type>
          <NewState>2</NewState>
          <CutType>5</CutType>
          <Conditions>
            <lecturer>calm,active</lecturer>
          </Conditions>
        </Possibility>
        <Possibility>
          <Number>3</Number>
          <Type>Time</Type>
          <NewState>5</NewState>
          <CutType>1,2,4</CutType>
          <Conditions>
            <slide>active,space,switch,annotate</slide>
            <lecturer>active</lecturer>
          </Conditions>
        </Possibility>
        <Possibility>
          <Number>4</Number>
          <Type>Time</Type>
          <NewState>6</NewState>
          <CutType>1,2</CutType>
          <Conditions>
            <slide>active,nospace,switch,annotate</slide>
          </Conditions>
        </Possibility>
      </Possibilities>
    </Time>
    <Event>
      <Possibilities>
        <Possibility>
          <Number>1</Number>
          <Type>End Of Lecture</Type>
          <NewState>15</NewState>

```

```

        <CutType>1</CutType>
        <Conditions>Empty</Conditions>
    </Possibility>
    <Possibility>
        <Number>2</Number>
        <Type>Question Acknowledged</Type>
        <NewState>7</NewState>
        <CutType>1</CutType>
        <Conditions>
            <questioner>acknowledged</questioner>
        </Conditions>
    </Possibility>
</Possibilities>
</Event>
</Transitions>
</State>
...
<State>
    <Number>15</Number>
    <Name>End</Name>
    <Context>0</Context>
    <Camera>LongShot</Camera>
    <CameraPiP>Empty</CameraPiP>
    <Transitions>
    </Transitions>
</State>

</Definition>
</FSM>

```

The definition of the FSM starts with setting the starting state. In this case, this state has the number 1. The states generally consist of their number, a describing name, the context assignment number and the names of the cameras realizing the shot. In contrast to the formal definition of FSMs we did not write down the transitions separately from the states and linked the both by an assignment but we listed all transitions going out of one state just below it.

The transitions are grouped into *time-based* and *event-based* transitions. With the exception of defining a time range, both groups have the same structure, but naturally a different content. This time range gives the lower and upper bounds out of which the actual duration is randomly determined. At first, the upper bound is set to a random value between the “max”-value *minus* the given time range and the “max”-value *plus* the given time range. Now, the lower bound is set to a random value between the “min”-value and the “recommend”-value. In the next step, a random value between the lower bound and the upper bound is chosen. As we can not foresee the activity in the lecture at a certain point of time, any value in this range is valid. If there is a reason to abbreviate or to extend the duration the FSM has to be defined in such a manner that either an event shortens the active shot or, due to e.g., activity in the shown

image, the same shot and therefore the same camera is chosen again with a new duration resulting in an extended time this camera is on air.

The reasons to do such an extensive effort is to become less predictive concerning the duration of a shot and to still assure that there is enough time to perceive all details, that the shot will not get too boring, and finally that the activity shown in an image will extend the duration while an activity not shown in this image will shorten the duration of this camera being on air.

Additionally, it is possible to influence the determined duration based on the conditions. For example, if a transition leads to a shot showing the slides and that the condition of a questioner is “inactive”, the determined duration can be extended by ten seconds giving the token “time+10” in the questioners’ condition entry. This takes into account that the slides are very important in transporting facts, and therefore it is a good idea to show the slides a little bit longer. As this very special token is only used in the question respectively in the answer context, it is obvious that it is only a tool for very special cases.

For further reference, please look at the definition of state 11 in Appendix 7.1.1. Theoretically, also the token “time-10” to shorten the duration by ten seconds and the token “time=10” to set the duration to exactly ten seconds is possible but they are not used in our prototype. The function “FSM.GetTimerInterval” (Appendix 7.2.1) implements this procedure and assures that the minimal duration is never less than four seconds.

All transitions of the XML file become “possible transitions” or short “possibilities” when loaded into memory. To be more precise, the FSM XML-description gets expanded while being loaded into the memory. Every transition in the file having more than one cut type will be represented in memory by separated possibilities having only one cut type each. So, we are able to determine a single transition by the probability calculation inside the main loop.

The main loop consists of many steps as shown here in pseudo code:

```

Start main loop
  If IncomingQueue.length>0
    Extract first EventMessage from Queue & eventually set EventSignal
    Extract all SensorInformation
  End If

```



```

If FadeOffTimerEvent
    Reset FadeOffTimerEvent
    Perform Part two of fading transition
End If
If AttentionSignal
    Reset AttentionSignal
    Collect all possible Transitions going out from the active state
    Calculate Possibilities & Return ResultingTransition
    Send ResultingTransition with Code "Attention"
End If
If EventSignal or TimeSignal
    Collect all possible Transitions going out from the active state
    Calculate Possibilities & Return ResultingTransition
    Send ResultingTransition with Code "Switch" or "Fade"
    Get Feedback of CameraStatus and CameraAlert
    If NOT CameraACK
        React on Status and/or Alert by adjusting Duration
    End If
    If there is still another event in the Queue
        Reduce Duration lengths
    End If
    Send SwitchCommand to AV Mixing Console
    If CutType is "Fade"
        Set and Start Timer for fading duration
    End If
    Update States History Log
    Set Active State, Context & Duration
    Reset Last SensorInputs & Last Events
End If
Execute OperatingSystem-Events
Sleep for 200ms
End loop

```

Within this main loop, the virtual director's core routine is the calculation of the probabilities for each possible transition and returning the resulting transition. We now go into detail of this routine (called *CalcPossibilities*).

```

CalcPossibilities
    Get State History
    Get Camera Status & MotionRate of every Camera
    Initialize possibilities & set probability to 100%
    Decrease probabilities according to how recently the new state of the
        transition was already shown
    Get the durations' basic constraints
    Only for time-based possibilities do

```

```

    Adjust probabilities according to their conditions in conjunction with
        the sensor inputs
    Add duration condition adjustments to its constraints
    Adjust probabilities based on the motion rates of each camera involved
        in the new state
    Adjust probabilities based on the camera status and camera alerts
End time-based possibilities
Only for event-based possibilities do
    According to signaled event set the according probability to 100%
    All other probabilities are set to zero
End event-based possibilities
Determine the highest probability value remaining
Determine all possibilities holding this highest probability value
If number of these max possibilities is larger than one
    Select randomly one possibility out of them
End If
Return the selected possibility
End CalcPossibilities

```

Inside this routine, the weight of all four adjustments to the time-based possibilities is equal at 25 percent: “States History”, “React on Sensors”, “React on Motion”, and “React on Camera”.

States History: The weight for the adjustment on the states’ history is basically determined by a linear function. In our prototype, we use a history buffer of five entries. The probabilities of transitions will be multiplied by 20 percent for transitions leading to states which have been shown most recently. Transitions to the two states least recently shown will be multiplied by 100 percent. In between, each buffer entry state will be multiplied with a percentage which is calculated with a linear function. The formula is defined as:

$$WeightOfHistoryBufferPosition = \min\left(\left(\frac{Position * (MaxPerc - MinPerc)}{(BuffLength - NumberMaxVals)}\right) + MinPerc, 1\right)$$

Definition/Formula 4: Formula to calculate weights of the history buffer.

We use the following parameters for our prototype:

```

BuffLength = 5
Position is zero-based (0 to BuffLength-1)
MaxPerc = 1 (100 %)
MinPerc = 0.2 (20 %)
NumberMaxVals = 2

```

For the five possible positions inside the states' history buffer, the resulting weights are shown in Table 3:

Table 3: Resulting weights of the positions inside the states' history buffer.

Position	0	1	2	3	4
Weight	0.2	0.46666667	0.73333333	1	1

At last, we provide a special treatment for zooms. Zooming is also represented as a transition, but it leads to the same state from which it originated. As we normally do not want to repeat a state and therefore a shot, we have to artificially “age” the state. So, for any transition marked with **SourceChange = False** its position used in the formula above is increased by one. However, in order to keep a differentiation between really “old” states and those which are artificially “aged”, we multiply the resulting percentage by the factor of 0.95, making it a little less likely.

React on Sensors: The next pass in calculating the probabilities focuses on the sensor inputs. These are marked with the name of the object they belong to e.g., lecturer or audience, etc. If the condition of a transition corresponds to this name, its input value is mapped to the transition. If more than one condition and value correspond to a transition their values are added. Then, the resulting value is used as a factor. If at least one condition signaled by a sensor matches a condition of a transition the probability of this transition gets increased by this factor. If no condition matches the probability gets decreased by 10 percent.

React on Motion: In this pass, the motion rate in the cameras' images is taken into account for calculating probabilities. The different cameras get different weights mapped to their motion rates: The “*LongShot*” of the lecture hall gets a weight of 21.5%, the “*Audience*” shot gets a weight of 23.5%, the “*Lecturer*” shot gets a weight of 25%, and finally the “*Slides*” shot gets a weight of 30%. These values have been determined by evaluating test series, but they can be adapted for other environments. If a camera is used as the source for the PiP part of the final shot, its value is divided by ten to decrease its overall influence to 10 percent while the value of the main shot is not changed. This is a little bit more than the proportion of the area the PiP camera uses in the whole image, which is 6.25%, in order to take the importance into account of the PiP shot has in comparison to the main shot.

React on Camera: At a first glance, it is confusing to have another routine reacting on camera inputs. However, in contrast to the previous values, this time the routine processes the status messages of the different cameras. Table 4 shows the factors mapped to the different cameras and their states.

Table 4: Mapping of camera status messages to factors.

	Lecturer	Slides	Audience	LongShot
Unknown/Error	0.1	0.1	0.1	0.1
AdjustingIris	0.8	---	0.8	0.8
Focusing	0.9	---	0.9	0.9
DetectingMotion	0.98	---	0.97	0.97
Moving	0.98	---	0.85	0.85
SearchingForPeople	0.98	---	0.97	0.97
Zooming	0.98	---	0.97	0.97
Idle	0.98	---	0.97	0.97
Else	0.1	0.99	0.1	0.1

For example, while the audience camera is first moving to, then focusing to, and finally zooming in on a questioners' position, all transition possibilities leading to a state using this camera get multiplied by the corresponding factors. In our example, the factors for using the audience camera are set subsequently to 0.85, to 0.9, and finally to 0.97. So, a shot using this camera gets more and more likely while its image gets more and more relevant. This enables the director to switch to this camera earlier, even if the shot is not yet fully established. Therefore, it gets more likely to show the image of the questioner in time when his or her question is audible. Again, the impact of a shot shown in PiP mode is set to 10 percent of its original value.

Not only time-based transition possibilities have to be taken into account. For event-based possibilities, only the event queue is relevant. Inside the *CalcPossibilities* routine, the queue is checked for the next event signaled, and then all possible transitions

are checked whether they do have a matching condition. If it matches, the corresponding transition possibility is set to 100%. If it does not match, it is set to 0 percent.

The events either refer to a certain state or to a context. If the event “*EndOfLecture*” appears, only the possibilities of transitions leading to the end state are set to 100%. If the event “*Questioner Acknowledged*” is given all possibilities of transitions leading to states inside the question context are set to 100%, etc. Besides the two events already shown, we check for the events called “*Lecturer AnswerIncomplete*”, “*Lecturer AnswerFalse*”, “*Questioner denied*”, “*Questioner deferred*”, “*Questioner stop*”, “*Lecturer Answering*”, “*Lecturer AnswerOK*”, “*OnlySlides*”, and “*NormalMode*”.

Among all (time- and event-based) probabilities the highest remaining value is determined and then all transitions whose probabilities equal this value are shortlisted. If there is more than one transition left, one transition is selected randomly out of the remaining ones.

After the transition is determined, its contained parameters have to be extracted, mapped to commands, and send to the various receivers. At first, the way of switching from one shot to another is extracted. Up to now, we have implemented the two possibilities “*Switch*” to describe a hard cut and “*Fade*” to describe a dissolve. Now, all involved cameras are determined: all cameras currently on air and all cameras which will be on air in the next shot.

In case of a hard cut, the present active cameras are switched to the off-air mode and the new cameras are put to the on-air mode. In case of a dissolve, all involved cameras are switched to the on-air mode at first, and a countdown timer is started. This second part of the dissolve is handled after this timer has expired and in the subsequent cycle of the main loop the cameras of the old shot are switched to the off-air mode.

Now, the duration of the new shot is determined. Then the cameras of the new state are requested, and if one of them reports a problem, the duration of the new shot is decreased in order to search for another shot as soon as possible. Then, the final duration is set.

At last, the according switch command is sent to the AV mixing console and recorder in order to put those cameras on-air whose on-air modes are set.

3.2. Cameraman Module

The cameraman module is very important for the Automatic Lecture Recording system. It was mainly designed and implemented by Manuel Benz in his Diploma Thesis (Benz, 2007). It was tailored to collaborate with the virtual director described above. We now present the details of the cameraman module.

The cameraman's obvious task is to shoot a high quality video, technically excellent and with an aesthetic image composition. This is easy for one person working alone, but if working in a team, it must be clear at any time which part to show and how to arrange the own image to make it fit into the sequence of shots of the other cameramen. So, a good communication within the team is necessary. The main challenge therefore is to turn the director's orders into useful and aesthetic shots immediately. Producing useful shots is based on the technical aspect of the job, and in our system it is implemented by image processing algorithms. However, as already mentioned, producing *aesthetic* shots must be based on cinematographic rules.

Some rules are already taken into account by accurately setting up the equipment, e.g., the line of action can not be crossed if the cameras' positions are chosen carefully, and also derived from the positions it is clear which camera has to frame him or her on the left side of the image and which camera has to frame a person on the right side of the image. However, in contrast to a human cameraman, the computer module is not able to overlook and to appraise the whole situation in the lecture hall and to properly react on complex situations. For example, the cameraman module can not detect whether a person in the audience wants to ask a question. Therefore, it is not able to zoom onto this person without getting an order from the director with the precise coordinates. In the next section we show the tasks the cameraman module has to fulfill in order to implement all three parts of a cameraman's job: the technical part, the aesthetic part, and the communicative part.

3.2.1. Tasks to Fulfill

The *technical part* is the one which can be done by the cameraman module itself without any help; it is the part controlling the camera properly. At first, the exposure has to be set correctly. The amount of photons a camera gets to take a picture or to take a shot is determined by three parameters: the aperture (iris), the shutter and the light

sensitivity of the film, respectively the sensor. If the amount of photons is too large the image or shot gets overexposed (too bright), and if it is too small it gets underexposed (too dark). As the sensor of the cameras we use has a fixed sensitivity and the maximum duration for each frame is 40 milliseconds because of the frame rate of 25 frames per second, the only parameter we are able to change is the value of the iris, the aperture.

Next, the cameraman module has to handle the focus. It can be set manually for our cameras, or their built-in auto focus can be used. An automatic operation for the zoom, for panning or for tilting, is not possible by the camera itself, but we will implement a routine to execute the orders coming from the director to point at a certain place in the lecture hall.

Also included in the technical part of a cameraman's job is the detection of the motion and its area inside the image. If a protagonist gesticulates a lot and therefore his or her hands get outside the shown image, the cameraman should detect this motion and distinguish it from the motion of a protagonist moving around.

At last, in the technical part, some additional background tasks have to be done, necessary as a base for the information the cameraman module provides for the director. These background tasks mainly consist of standard routines of image processing, like creating differential images, histograms, noise filtering, motion ghost elimination, skin detection, person detection, region growing, region combining, and motion rate determination. It is important for our approach that every image processing algorithm can be executed in real time as our system is used for live production. This leads to trade-offs between accuracy and speed. We decided to use an algorithm with sufficient accuracy while running in real time.

The *aesthetic part* of the cameraman module realizes functions like

- a) how to frame a person in an image, whether he or she should be shown on the left, on the right, or in the middle of the image;
- b) to zoom out when a gesticulating protagonist has been detected by the technical part in order to show the action completely;
- c) to follow a moving person when detected by the technical part;
- d) to move the camera at different speeds depending on its on or off air status.

The *communicative part* is necessary to work in a team. It consists of a bidirectional communication between the cameraman and the director. While in real life a bidirectional communication among all participants of a live production is realized by the intercom system, it does not give any advantages if e.g., the virtual cameramen can communicate with each other. In our system, each cameraman module receives the orders from the director module and is able to give back a status report, respectively a feedback, concerning the camera status and its own status.

3.2.2. Implementation Details

We do not describe all the details of the cameraman module but choose to go into some details which are especially relevant for the whole system, or if they have been improved since the Diploma Thesis of Manuel Benz (Benz, 2007).

A crucial point is the determination of a correct exposure for the image. As explained above we only have left the iris setting to adjust the exposure. The basic situation in lecture halls is a well-lit room, so in most cases it is only necessary to find the correct iris setting. The Tables 5 and 6 show f values, their corresponding ratios of incoming light, and their camera parameter values.

Table 5: Iris control: f values, ratio of light and camera parameters; part 1.

f value	1.0	1.2	1.4	1.7	2.0	2.4	2.8	3.4	4.0	4.8	5.6
Ratio of incoming light	1.0000	0.7249	1/2	0.3192	1/4	0.182	1/8	0.0794	1/16	0.0457	1/32
Camera parameter value	9998	8332	7141	5881	4999	4166	3571	2941	2500	2083	1785

Table 6: Iris control: f values, ratio of light and camera parameters; part 2.

f value	6.8	8.0	9.5	11.0	13.5	16.0	19.0	22.0	27.0	32.0
Ratio of incoming light	0.01980	1/64	0.01149	1/128	0.00476	1/256	0.00202	1/512	0.00112	1/1024
Camera parameter value	1470	1250	1052	909	741	625	526	454	370	312

The ratios of incoming light are relative to the open iris (f value 1). The tables are based on the table of (Millerson, 1990) on page 25 concerning f values and incoming light ratios and on the corresponding given camera parameters. They can be calculated by Formula 5, based on the minimum (1) and maximum (9999) camera parameter values:

$$\text{CameraParameter}(f - \text{number}) = \frac{(9999 - 1)}{f - \text{number}}$$

Definition/Formula 5: CameraParameter based on f -number of the iris.

In the iris initialization phase, all possible iris values are tested, and the resulting image is evaluated. As we focus on a good luminance distribution inside the image, it is enough to evaluate a black and white, or to be more precise, a gray scale version of the image. As everything should be done in real-time, we looked for a robust but fast measure to evaluate the exposure. For our task, the arithmetic mean of the histogram is sufficient. Valid f-numbers are found by comparing the arithmetic mean with a given minimum and a given maximum out of the cameraman's configuration file. Out of all valid f-numbers, the median is selected as a result of the initialization phase.

During the standard operation of the camera, the iris setting is adjusted automatically as long as one of the valid f-numbers is sufficient to re-achieve a well-lit image. If the lighting conditions change extensively and no valid f-number can be used to compensate this change, the iris has to be re-initialized. If the camera is on air when such a drastic iris mismatch is detected the cameraman reports a camera alert to the director before starting the re-initialization phase.

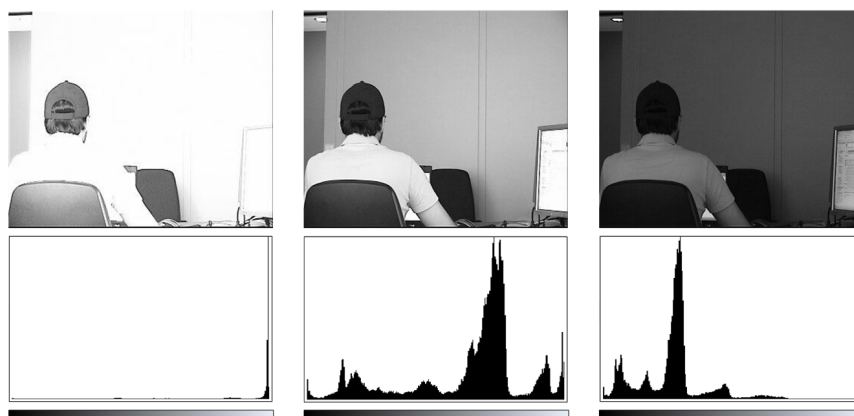


Figure 11: Three exemplary exposures (Benz, 2007).

Figure 11 shows typical examples of an over-exposure (on the left), an under-exposure (on the right), and a correct exposure (in the middle), together with their histograms.

Besides this standard iris setting procedure and its continuous adjustment, we have implemented a special routine to compensate backlight situations when showing a person. This is useful both for the lecturer and for a questioner out of the audience. In both cases the person is the important part of the image. In contrast to the standard solution where the entire image is examined, now only the person is considered. We use skin color detection and join the detected regions to define the relevant parts of

the image. The iris setting procedure is then applied only to these regions. On the right side, Figure 12 shows the result of the skin detection algorithm applied to the original image (left). The false positives from the background can be eliminated, e.g., by background subtraction.



Figure 12: Skin color detection example (Benz, 2007).

The background model is created out of two consecutive images shown in Figure 13, images (a) and (c). Its result is shown in image (b), which is a bit fuzzy as the person has moved only a little. Image (d) shows the differential image, making the motion obvious.

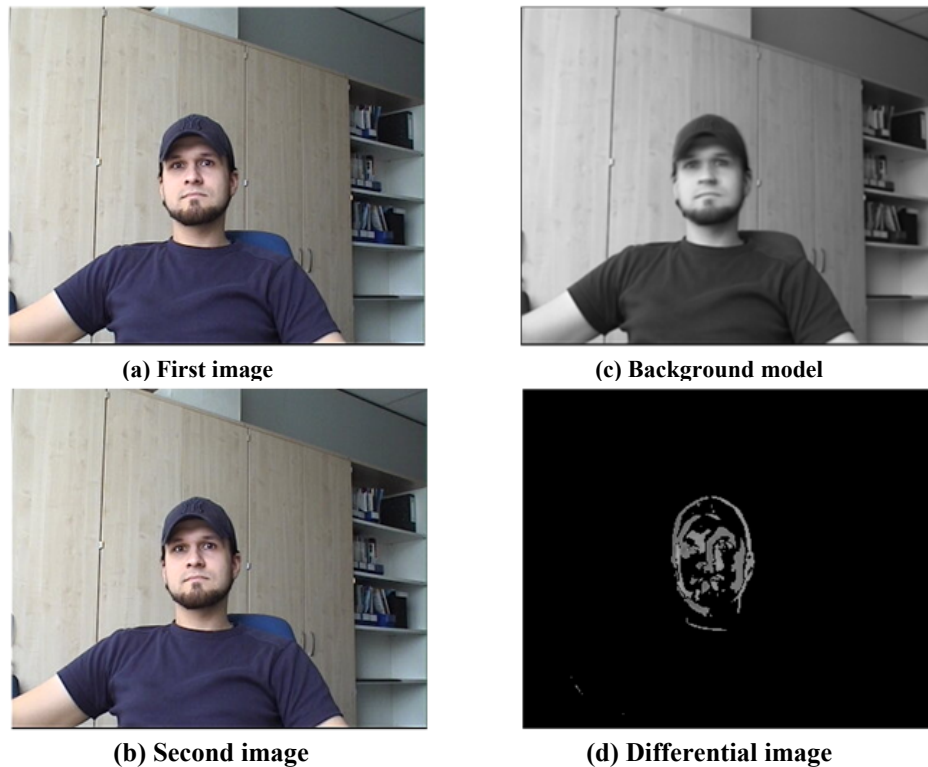


Figure 13: Images of the steps in background subtraction (Benz, 2007).

The background model can be created out of more than two consecutive images in order to identify very slow motions, too. So, we can differentiate between regions not moving at all and regions with even a little motion. In combination with the skin color detection algorithm and the joining of adjacent regions, we are able to precisely determine a person in an image. Figure 14 shows an example of this combination of routines, marking the segments found and joining the adjacent regions.

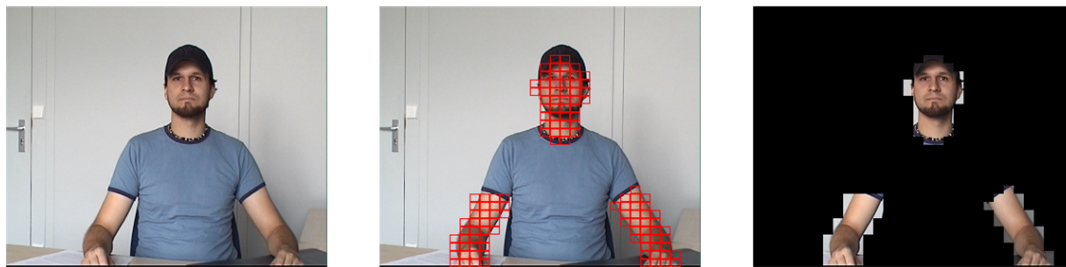


Figure 14: Skin detection and region joining example (Benz, 2007).

Finally, we apply the iris setting algorithm to the joined skin colored regions. The result of the standard automatic backlight compensation algorithm (left) and the result of our approach (right) are shown in Figure 15.



Figure 15: Results of two backlight compensation algorithms (Benz, 2007).

So, we use the standard algorithm to set the iris in most cases, but if we focus, e.g., on a questioner, we use our approach with skin color detection.

Starting from the cameraman module version of the Diploma Thesis, we improved some parts. A very important example is the motion rate determination in an image. In the case of processing the image of the slides camera, we have changed the original routine. The reason was not a problem of the original algorithm but a special case not occurring in normal images. Like most routines of the cameraman module, the motion detection and motion rate determination works on gray scale images to reduce the workload. For standard camera images, this approach works very well, but for the

slides we encountered a problem from time to time: as long as the lecturer writes on a new slide, adds comments to slides, or even changes the slide, the routine works correctly. But if the lecturer explains, e.g., a diagram and traces the lines in order to show which part he or she is referring to, only the color of the line changes but no or only a little change of the gray scale image can be found. The example in Figure 16 makes this obvious. It shows the same image, on the left side as a color image and on the right side as a gray scale image.

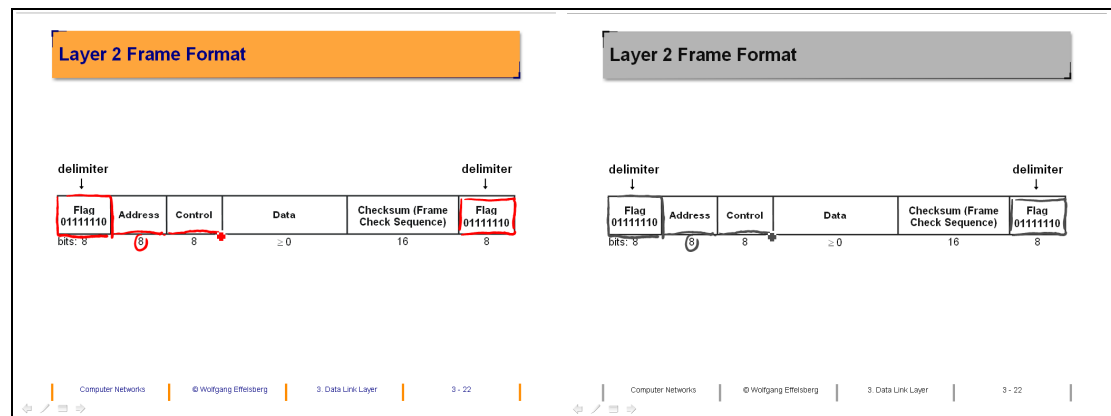


Figure 16: Comparing annotation visibility of color and gray scale images.

The reason why this effect does not occur in normal camera images is that it is very seldom that an object only changes its color without changing its shape or without moving at all. So, we amended the cameraman module with a routine to detect motion and to determine the motion rate in colored images.

For the gray scale images, we had to do some changes in order to achieve a useful differential image. In order to get rid of the sensor noise of the camera, we need to apply a bandpass filter, and we need more than two images to create an average in order to find slow moving parts in the image while avoiding them appearing as “ghosts”. The effort preparing the images of the slides camera is less as we do not use a real camera but a scan converter for the screen in conjunction with a streaming server to capture the VGA signal and create the same MPEG-4 and MJPEG streams as from the real cameras. As no image sensor is used, the noise in the images is less, and so we do not need the bandpass filter. As we want to react on any change in a frame, we do not need the average over more than two images: instead, we take two neighboring frames and subtract them from each other. We use color images, and the differential image also contains colors. We extrapolated the way of subtracting gray

scale images: we do not subtract one-dimensional values but three-dimensional vectors of the color space. This differential image gets segmented, and the segments are analyzed. For each pixel in the segment the distances in the color space to its eight surrounding neighbors are calculated and added up. If this cumulated distance exceeds a threshold, the according segment gets marked. The ratio of marked segments to all segments describes the motion level of the image, and as the marked segments are known, the areas in which motion appears are also known.

We measured the scan converter noise, and therefore set the threshold of the cumulated distance to 50 units in color space. It is obvious that this value is not too high if we have a closer look on the average distance each neighboring pixel may theoretically have: a threshold of 50 units divided by eight neighboring pixels equals to 6.25 units as the average distance per pixel while the highest possible distance value is $\sqrt{3 \times (255 - 0)^2} = 441.6729559$ units. It is a kind of low-pass-filter which eliminates the noise of the camera sensor and still reacts precisely on real changes from frame to frame. The distance per pixel is the length of the difference vector of two points in the RGB color space, as shown in Figure 17 and in Formula 6.

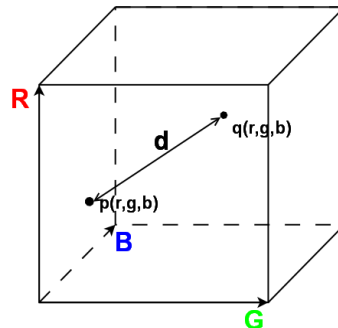


Figure 17: Distance of two points in the RGB color space.

$$d = \sqrt{(r_p - r_q)^2 + (g_p - g_q)^2 + (b_p - b_q)^2}$$

Definition/Formula 6: Length of a 3D vector; here the distance between the two points named p and q.

Some important parts of the aesthetic component of the cameraman module are routines to arrange the image in a similar way as a real cameraman would do. We will show three aspects and their implementation in the cameraman module. At first, there should be a difference in moving the camera whether it is on air or not. Second, depending on the director's demand, it is necessary to arrange a person on the left side,

in the middle, or on the right side of the image: this is an important prerequisite to create a shot – counter-shot scenario. Third, a cameraman’s reaction on a heavily gesticulating protagonist: he or she will try to zoom out at first to get all the action into the image as it produces a steadier shot (rather than panning or tilting the camera).

While a cameraman moves the camera as fast as possible to the next shot when he or she is off air, it is important to move the camera in a smooth and slow way during a shot. An experienced cameraman even accelerates when moving the camera up to a certain speed for panning and slows down the speed before stopping the pan. The cameras we use neither support accelerating nor slowing down, so we tested different fixed speeds. Finally, we set the camera’s speed to 100 percent if the camera is switched off air and to 25 percent if it is on air. So, we achieve a quick adjustment in the off-air mode and a smooth panning in the on-air mode.

Besides determining the correct exposure setting, the skin color detection is used in combination with a face area estimation routine is used for person detection. For all areas in which skin color is detected their center of gravity is calculated. It is a good assumption that the head is located above the arms, so we will use the uppermost detected area as the face area if more than one area is detected. For the camera of the lecturer and the camera of the questioner, i.e., the camera of the audience in the questioner mode, it is a good idea to put the protagonists on opposite sides of the image to be prepared for a shot – counter-shot scenario. The cameraman module therefore needs some specifications: the coordinates of the camera in the lecture hall, on which side of the line of the lecture hall the camera is positioned, and the coordinates of the current protagonist in the lecture hall: with these parameters it is possible to arrange the frame so that the protagonist is set to one side of the image, and the line of action is not crossed. Naturally, the other cameraman module will behave accordingly, and will put its protagonist on the opposite side of the image. Figure 18 shows the test results of putting a protagonist, in this case an abstracted protagonist, on the left side, in the middle, or on the right side.

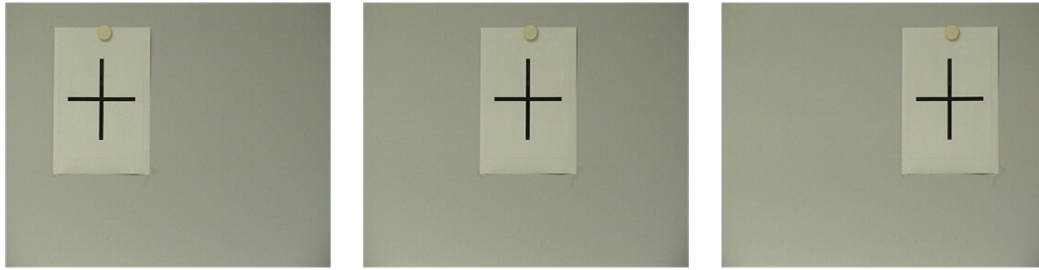


Figure 18: Frame arrangement tests of abstracted protagonist (Benz, 2007).

Figure 19 shows exemplary shots of a lecturer and a questioner.

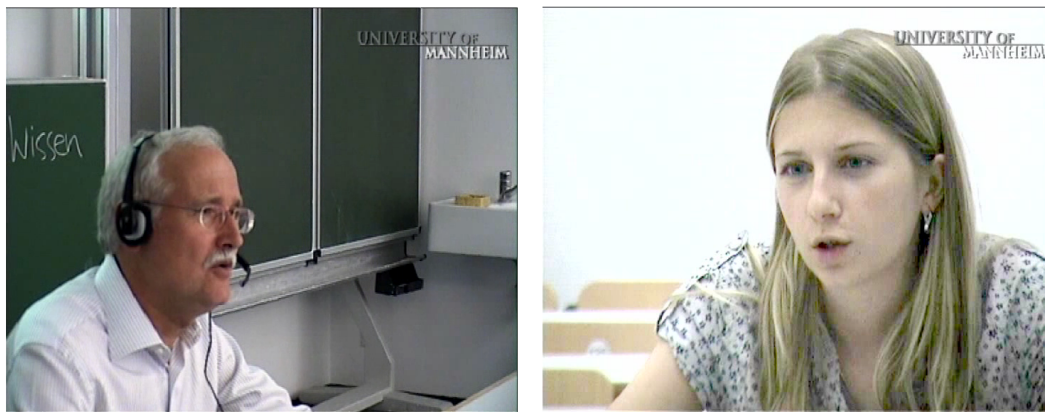


Figure 19: Lecturer and Questioner aligned in a shot - counter-shot scenario.

Very typical for many automated cameramen are the permanent small panning movements if the protagonist is weaving his or her upper part of the body and therefore the chosen image is “too small” showing him or her. A human cameraman will simply zoom out a bit to enlarge the area shown. We avoid this unpleasant mechanism by a special provision: when the protagonist is shown too close there is a lot of motion in the image while the protagonist is moving or gesticulating. So, if the motion level of an image is above a given threshold, we give the order to the camera to zoom out a bit. Now, we re-check whether the motion in the new picture is below the threshold. As long as the motion is above the threshold, we keep zooming out slowly. If the motion level is below the threshold for a certain amount of time, we start to zoom in again until the original zoom level is reached again. The complete process is shown in the seven images of Figure 20 from left to right.



Figure 20: Motion-triggered automatic zoom-out followed by automatic zoom-in (Benz, 2007).

For those cases in which the protagonist really moves, we check after zooming out whether his or her face is still moving, and whether it is too far out of the place we want it to be, which can be the left, the right, or the middle of the image. Only if these two requirements are given, we start to adjust the camera by panning and following the protagonist.

The communicative part of the cameraman module mainly deals with the reception of orders from the director module and giving feedback to it in form of status reports, including status alerts. It is the abstracted form of the communication done between humans via intercom during a live production. We decided to realize a two-tier approach for the cameraman in order to make it adaptable to different camera models by only changing the interfaces to the new camera. Figure 21 shows the complete information exchange among all three components involved.

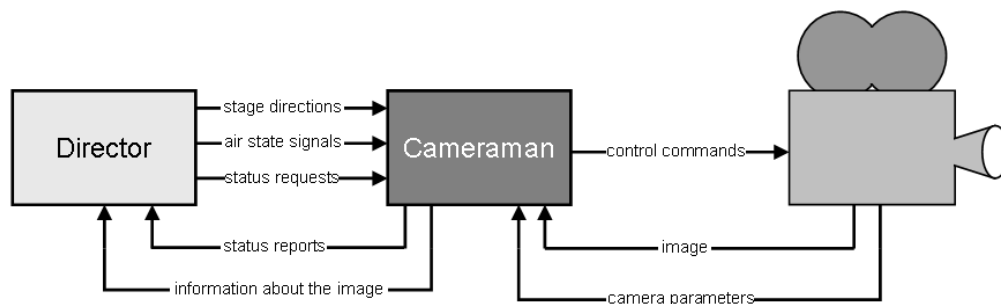


Figure 21: Information exchange from director to camera via the cameraman and back (based on Benz, 2007).

The communication between the director module and the cameraman module is done by Extensible Markup Language (XML) messages via TCP/IP. Besides the easiness to parse such messages, it allows to easily wrap an arbitrary number of parameters and

values in a predefined scheme. For example, it is very easy to put only the requested values in a status report without changing the scheme. In addition, XML messages are human-readable, and therefore the messages can be checked easily, e.g., during debugging.

The stage directions, the air state signals and the status requests the director module sends are very short commands, still in XML style as the examples in Table 7 show.

Table 7: Exemplary stage directions from the director module to the cameraman module.

Command	Description
<code><CAMERA_PROPERTIES/></code>	Requests all camera properties during initialization process.
<code><STATUS_REQUEST/></code>	Requests all current camera and cameraman parameters. Possible at any time.
<code><ON_AIR_SIGNAL/></code>	Sets the camera and the cameraman into on-air-mode.
<code><OFF_AIR_SIGNAL/></code>	Sets the camera and the cameraman into off-air-mode.
<code><LECTURE_END_SIGNAL/></code>	Signals the end of the production, resets all parameters of the cameraman and of the camera.
<code><MOVE_LOCAL X='x.xx' Y='y.yy' Z='z.zz' /></code>	Sets the new coordinates on which the camera should point at.
<code><MOVE_HOME/></code>	Returns the camera into its home position.

The status report as the answer to the command *status request* of the cameraman only consists of one parameter whose values are:

`Unknown/Error=0, Idle=1, Moving=2, Zooming=3, Focusing=4, AdjustingIris=5, DetectingMotion=6, SearchingForPeople=7.`

This parameter gets encapsulated in the status request reply, here is an example:

`<STATUS_REQUEST>1</STATUS_REQUEST>`

The motion report as the main information about the image is accompanied by the camera status value, which enables us to differentiate whether a motion level comes from the camera action or from the image itself. Here is an example:

`<MOTION_REPORT LEVEL='17' STATUS='3' />`

The reply on the requested camera properties during the initialization phase is a little bit more complex:

```
<CAMERA_PROPERTIES>
  <properties>
    <cameramannname>Axis214</cameramannname>
    <cameramanipaddress>134.155.92.33</cameramanipaddress>
    <camerattype>axis</camerattype>
    <control>ptz</control>
    <cameratarget>audience</cameratarget>
    <streamingaddress>134.155.92.12</streamingaddress>
  </properties>
</CAMERA_PROPERTIES>
```

The cameraman module translates the directions into commands for the camera itself. The effective translation depends on the model of the camera and its interface. In order to cope with different models and interfaces of cameras we realized a two-tier approach. This enables us to easily exchange only the hardware dependent tier without rewriting the whole cameraman. Our prototype uses cameras and video servers from the same manufacturer. This leads to a unified interface for all commands. The basic structure of these commands is an URL containing all parameters:

```
http://134.155.92.23/axis-cgi/com/ptz.cgi?<parameter>=<value>[&<parameter>=<value>]
```

The parameters and some of the possible values are listed in Table 8:

Table 8: Parameters and possible values of the camera interface.

Parameters	Possible values
autofocus	"on" "off"
autoiris	"on" "off"
center	?x,y [pixel,pixel]
focus	value [0-9999]
imageheight	value [pixel]
imagewidth	value [pixel]
iris	value [0-9999]
pan	angle-value [-180 - 180]
speed	value [1-100]
tilt	angle-value [-180 - 180]
zoom	value [0-9999]
query	"speed" "position"

In the last row, the parameter to query the camera parameters is shown. Querying the speed returns an HTTP response containing the single value of the actual speed set,

but when querying the position, the HTTP response contains a lot more information. An example of such a response can be found below:

```
HTTP/1.0 200 OK
Content-Type: text/plain

pan=-50
tilt=-3
zoom=500
focus=600
iris=800
autofocus=on
autoiris=off
```

Similar to the way of placing steering commands and querying the camera parameters is the way of requesting images out of the MJPEG stream of the camera. Here an exemplary call:

```
http://134.155.92.23/axis-cgi/mjpg/video.cgi?resolution=352x288
```

The HTTP response contains the JPEG data of at least one JPEG image. The format of the response is shown in the following example:

```
HTTP/1.0 200 OK
Content-Type: multipart/x-mixed-replace;boundary=myboundary

--myboundary
Content-Type: image/jpeg
Content-Length: 15656

<JPEG binary image data>
```

So, we get a JPEG-coded bitmap of the camera image, which we are now able to analyze in the cameraman module.

Not shown in Figure 21 is the video stream which is recorded by the AV mixing console / recorder as it will be described in detail later. The stream which is used for recording is an MPEG-4 video stream which is produced in parallel to the MJPEG stream used for our image processing. Thus, it is possible to do both tasks without interference.

3.3. Sensor Tools Module

The sensor tools module is designed to provide one consistent interface for arbitrary sensors. We defined a queue of type STRING to transport sensor information to the virtual director to make it possible to transport messages, including optional parame-

ters, and parse them. If further sensors are needed, only the parsing of the new messages has to be implemented in the corresponding routine inside the virtual director. The messages of the virtual cameramen are also queued in the same queue, which makes the virtual cameraman a special version of a sensor. There are two types of cameraman messages: sensor data, e.g., motion reports, and camera alerts, e.g., “image too dark”. This is a very comfortable way of integrating the data of the virtual cameramen into the processes of the virtual director.

The procedure done by the virtual director is to check at first whether any messages are queued, then all queued messages are parsed by their type in the following order: *alert*, *event*, *information*. While queued messages of the type *alert* need immediate processing, messages of the type *events* are processed at the end of the next shot. Messages of the type *information* are considered during the normal calculation of possibilities while determining the next transition. For each type of messages, the processing order of the queued messages is first in – first out (FIFO).

In addition to the order of processing the different queued messages, the queue itself can be manipulated based on the message currently processed. The most obvious case is when the “*EndOfLecture*” event message is processed, the complete message queue is cleared as no further messages need to be processed after ending this run of the virtual director. Similar messages which cause a change of the director’s context exist: for example, if a message is changing the current context into the questioner’s context, all messages still in the queue requiring the same context change are purged out of the queue. Thus, we make sure to react only once on an event which may be registered by multiple sensors, and keep the message queue short.

3.3.1. WLAN Indoor Positioning System

As we need to know where a questioner is located in the audience, we come back to the indoor positioning system based on wireless LANs developed by Thomas King at our institute (King *et al.*, 2007). It uses the already deployed access points used for the mobile access infrastructure at the University of Mannheim, and can be amended with additional fixed access points for better accuracy. The most accurate position estimations are achieved if four to five access points are used according to (King *et al.*, 2007). As his implementation was done in JAVA and our system is based on the Microsoft Visual Studio suite, his software had to be ported in order to be compatible.

We appreciate that this work was done by Hendrik Lemelson as described in (Lampi *et al.*, 2009).

Tasks to Fulfill

The WLAN indoor positioning system runs as daemon providing its service on the PDAs used by the students during the lecture. This includes that it automatically starts after the PDA has been switched on, estimates the position independently, and finally preserves the estimated position and provides it any time it gets requested. Additionally, it should automatically refresh the estimated position after a certain time range.

Although we describe this software under the sensor tools module it does not use the message queue itself. Instead, it provides auxiliary data for other sensor tools, especially the Question Management software. The estimated position gets requested and transmitted to the virtual director module, coupled with the messages of the Question Management software. It makes them complete and meaningful.

As the sensor tools module should provide the director module with complete information at once, it is useful to send a message like “*Question from seat 98*”. Otherwise, a bidirectional communication channel must be established, which at first transports a message like “*Question announced*”, requests back something like “*where from*”, and then gets a second answer like “*Seat 98*”. To avoid such a complicated communication, we decided to assemble a message out of all the necessary information before sending it.

Implementation Details

The WLAN indoor positioning system is a two-phase system: It consists of a *training phase* and a *position determination phase*. During the training phase, the characteristic parameters of the available WLAN access points are recorded in a certain grid, in our scenario on every seat of the audience in the lecture hall. In order to get a meaningful, so-called “fingerprint” of the WLAN environment, 110 measures are taken at every measuring point. A fingerprint contains the signal strength of all the access points received. All fingerprints are stored in a database which gets deployed on any PDA used by the students in the second phase, the position determination phase. The PDA then measures the current signal strengths of the access points in reach and compares the measured results with the ones stored in the fingerprint database. As it is very unlikely to measure an exact match for a parameter, the comparison uses statistic

methods to determine how likely a certain position is. The most likely position is returned as the result.

The position estimation is realized as a service running on the PDAs automatically. In order to save energy, it is done once the PDA is started, and it only gets repeated if the last measurement is older than one hour, or a new measurement is manually triggered. Once the service has determined its position, it returns the result at any time requested. This result is used without any further processing by the Question Management software described in the next section.

As position estimation based on 802.11 (WLAN) fingerprints offers a positioning accuracy of approximately two meters on the average. This accuracy is sufficient for many applications but is not optimal for our case because it covers up to three seats in a row. So, up to nine students can have taken their seats in the circumference of about two by two meters.

We improve this coarse result by two measures: at first, based on the estimated position, we show an area of three by three seats to the user and let him or her specify the exact position by clicking on the number of the seat on the GUI of the questioner's client application on the PDA, as shown in Figure 22.

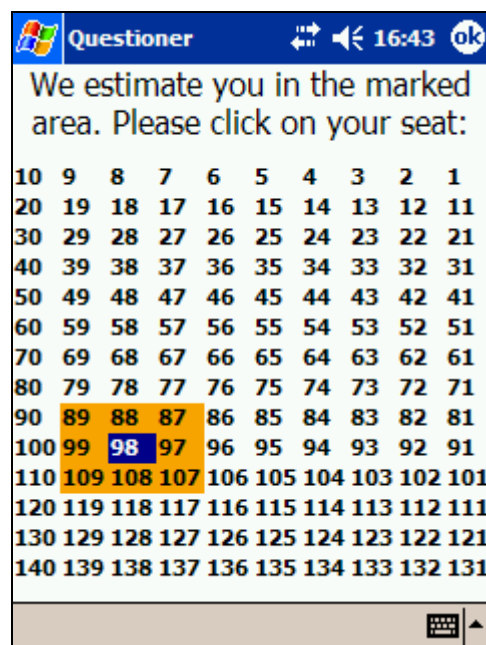


Figure 22: Estimated area marked and seat of student confirmed.

The second measure we take is to choose a wider framing by the cameraman. We zoom into the audience until up to three persons are visible instead of zooming onto

one person. The cameraman module itself is able to refine the zoom and the adjustment based on its person and motion detection algorithms afterwards.

3.3.2. Question Management Software

The Question Management (QM) software is necessary to keep track of the lecturer-audience interaction as the virtual cameraman is able to detect motion in an image but is not able to detect the semantic meaning of this motion.

The software consists of three modules. The abstracted view on the participants of the interaction leads us to the first two modules: the lecturer's client and the questioner's client. As we need a central communication base which also keeps track of the active state of the interaction we amend our modules by a communication server which synchronizes the lecturer and questioner clients and produces the messages for the virtual director.

This makes the QM software a distributed software suite. As the lecturer already uses a computer to present his or her slides and another computer is used for the virtual director, we decided to use these machines to run the corresponding part of the software suite on them. In order to equip the potential questioners in the room with electronic devices we use PDAs which we employ anyway for interactive quizzes during the lecture (Scheele *et al.*, 2005) and (Kopf & Effelsberg, 2007). They provide wireless LAN access, a built-in microphone and a touch screen. These three features are used to implement the questioner's client.

Tasks to fulfill

The QM software suite has to provide the GUIs for the lecturer and the questioner, it has to implement and supervise the lecturer – questioner interaction as shown in section 2.2.3, it has to convert the actions of lecturer and questioner into messages for the sensor input of the virtual director, and finally it handles the audio connection of the questioner for the virtual sound engineer. Furthermore, the software may have to manage more than one hand-raising at the same time, and it additionally has to provide a management interface for the lecturer.

Implementation Details

We now show the important implementation details of the three modules of the Question Management software suite.

Questioner's Client

The first purpose of the Questioner's Client is to request the estimated position from the WLAN indoor positioning system, to provide the GUI to show the estimated position and let the user confirm his or her seat. Then the client connects to the QM server, transmitting its IP address and the confirmed position.

The second purpose of the questioner's client is to provide the general GUI for the questioner. In the top portion of the screen, there is a status indicator box. It informs the questioner at any time about the status of the interaction. The status indicator changes according to the interaction from “*waiting*”, “*announced*”, over “*please ask*” to “*answering*”. Figure 23 shows the standard interface.

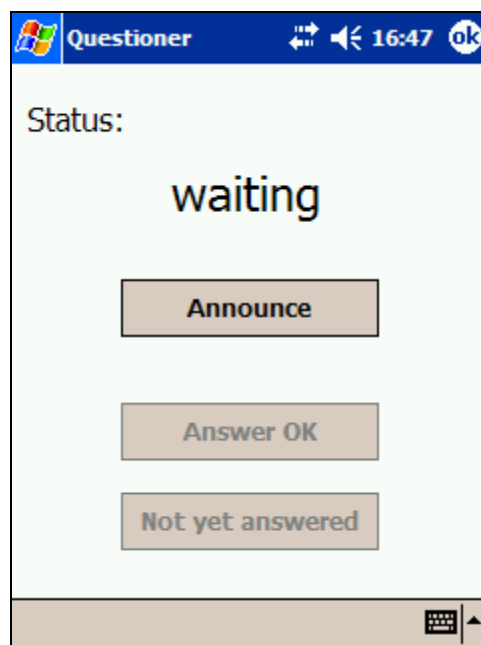


Figure 23: Standard interface of the questioners' client.

However, in some cases, there are different messages, e.g., “*sorry, not possible*” if the question was blocked by the server, “*just a minute*” if it was deferred by the lecturer who will automatically be reminded after one minute, or “*please, do listen*” if the lecturer denies the question.

The third purpose of the client is to capture the student's audio and transfer it to the server in order to get it processed by the sound engineer. As the PDAs are typically within one's arm reach, they are in a distance well suited for audio recordings. The audio transmitting instance is created when the questioner announces a question, but

the transfer of the audio data using wireless LAN is not started until the questioner has been given the floor.

Lecturer's Client

This client runs on the lecturer's presentation computer. Its central task is to indicate that a student would like to ask a question, and to provide an intuitive user interface to accept or decline such questions. At a certain point in time, a question may disturb the progress of a lecture, or it can be undesirable for other didactic reasons. Therefore, the client also offers the possibility to temporally postpone a question.

While teaching, it is important that the client software of the question manager does not disturb the lecturer too much. He or she should only focus on the QM client in case of a question. Therefore, we have implemented the client in such a way that it is usually running in the background. If a student wants to ask a question the questioner's client submits this request to the lecturer's client via the QM server. A foreground window pops up at the lecturer's computer, stopping the current presentation. If more than one student announces a question, the foreground window shows a list of all announced questions. Figure 24 shows the foreground window with one announced question.

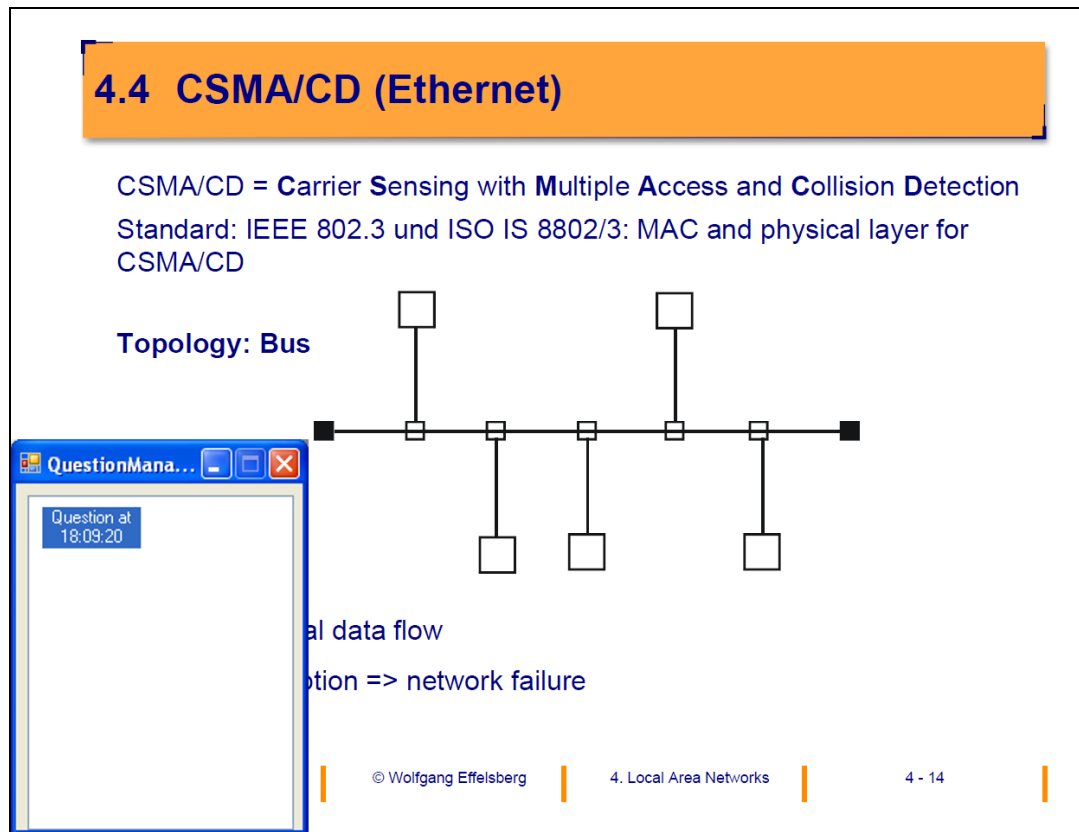


Figure 24: Popup on the Lecturer's computer showing announced questions.

A mouse click on a button of the lecturer's client is sufficient to give the floor to the student, to ignore the request, or to postpone it. In the last case, the window pops up again after the predefined time of one minute. Depending on the lecturer's decision how to proceed with the question, the student gets a different status messages on the display of his or her PDA, as mentioned in the section on the QM questioner's client above.

Figure 25 shows the steps of the basic question – answer interaction, amended with the corresponding client screen shots of the questioner (left) and of the lecturer (right).

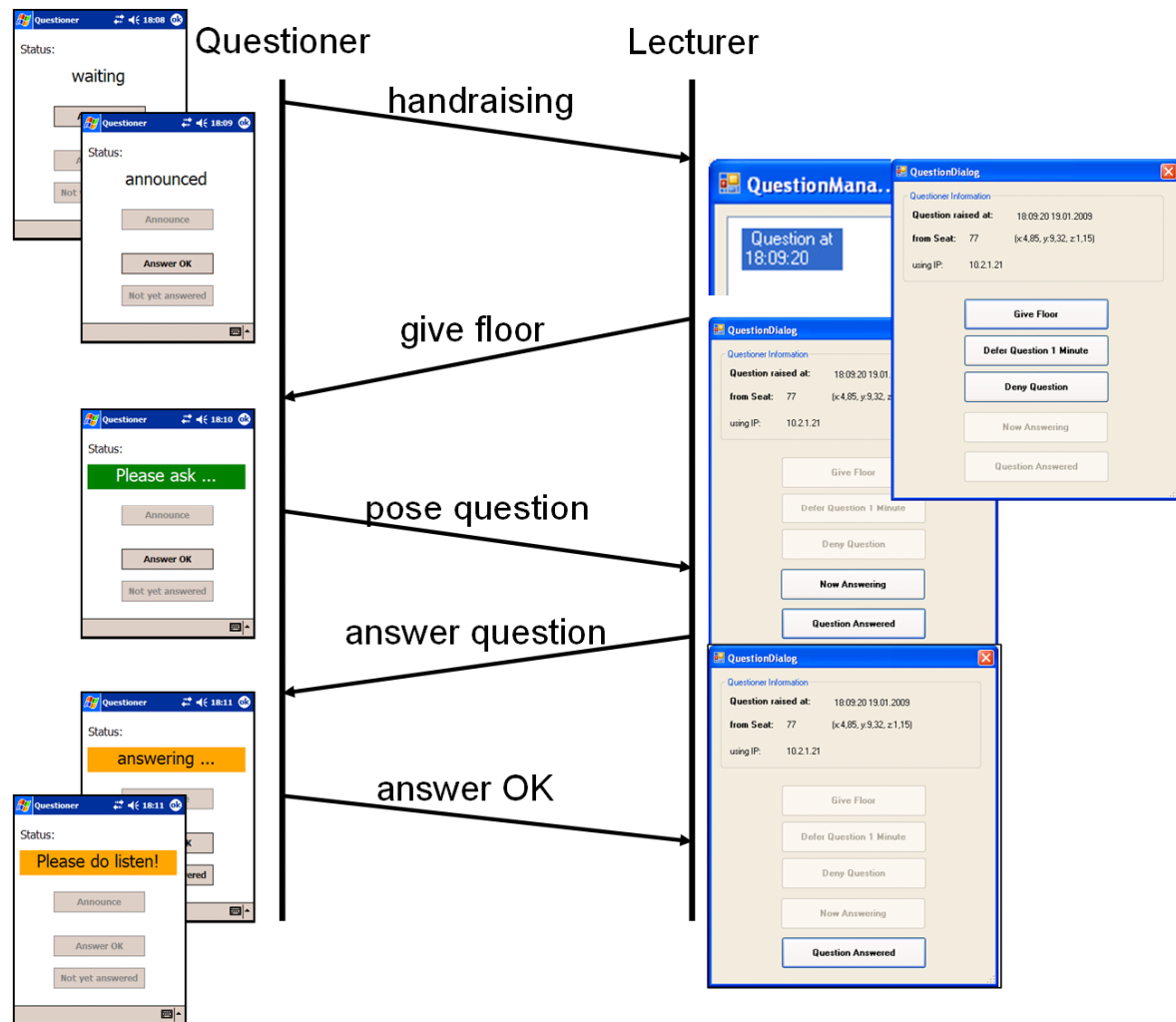


Figure 25: Basic question - answer interaction, amended with client screen shots.

After the lecturer has given the floor, the student asks his or her question, and the QM client transports his or her audio to the sound engineer. One click on the button “*Now answering*” in the lecturer’s client indicates the beginning of the lecturer’s answer. A last click by the lecturer on the button “*Question Answered*” signals the end of this question – answer interaction. The lecturer may allow additional questions, remarks, or discussions, but he or she can also continue with the lecture.

Transporting the questioner’s audio to the sound engineer is necessary for the recording of the lecture and for tele-teaching scenarios as the remote audience or viewers of the resulting video want to hear the questions. In the lecture hall itself the voice of the questioner is normally sufficient for every participant.

Not shown in the basic version of the question – answer interaction above is the possibility for the questioner to ask a further question in the same interaction. The GUI of

the QM questioner's client has two buttons which are active when the lecturer is answering. So, the questioner can either click on the "*Answer OK*" button which ends the interaction or hit the "*Not yet answered*" button. In the latter case, the lecturer's GUI is reset to the state before he or she gave the floor to the student, and there is room for another turn.

A question – answer interaction is active until either the questioner has hit the "*Answer OK*" button or the lecturer has clicked on the "*Question Answered*" button on their respective GUI. Of course, the decision of the lecturer overrides the action of the questioner.

Question Management Server

Let us finally look at the *QM Server* application which represents the central component of the Question Management software suite. It handles the communication with all the questioner clients concerning all interaction events, of receiving the questioner's audio when given the floor, of communicating with the lecturer client for the relevant interaction events, sending consolidated events to the virtual director as sensor input, and providing a GUI to monitor and control the client events.

During start-up, every QM questioner client automatically registers itself at the QM server by sending its IP address and its position coordinates. The connected QM questioner client is represented by switching the background color of the according field of the server GUI to green.

When a questioner announces a question the server receives the corresponding event message and performs three actions:

1. Create an instance of the audio receiver for the questioner client,
2. Raise an event for the lecturer client, and
3. Raise a sensor input event for the virtual director.

While a window pops up at the lecturer's client, the virtual director gives the orders to the cameraman showing the audience to aim at the position coordinates of the questioner and to zoom in. When the lecturer gives the floor to the questioner, the server receives the corresponding event from the lecturer's client. Now, the server sends out the sensor input event "*questioner acknowledged*" to the director, and as a result the audience camera is switched on air. Meanwhile, the audio stream is sent from the

PDA to the QM server over wireless LAN, and the questioner is requested to ask his or her question.

When getting the signal from the lecturer's client that the lecturer starts to answer the question, the server stops receiving the questioner's audio and announces that the answer begins to the questioner's client as well as to the virtual director. At last, when the server gets signaled the end of the answer either from the questioner or from the lecturer, it resets all displays settings and audio components and reports the fact that the system should be switched back to the lecture context by the virtual director.

Furthermore, the server provides some more functionality:

- It blocks any question announcement if the "*Block Clients*" property is ticked.
- By clicking on a number of the lecture hall abstraction shown in the GUI, the audience camera is aimed at the respective seat.
- Finally, the "*Home Position*" button resets the camera adjustment to show the entire audience.

Figure 26 shows the GUI of the server. Registered questioner clients are shown in green while the currently speaking questioner is shown in red. On the left side, the queue of announcing students is visible, showing each with the timestamp and the seat of the announcement. On the right side, there is some additional information concerning the IP address of the active questioner and the state of each part of the QM software suite.

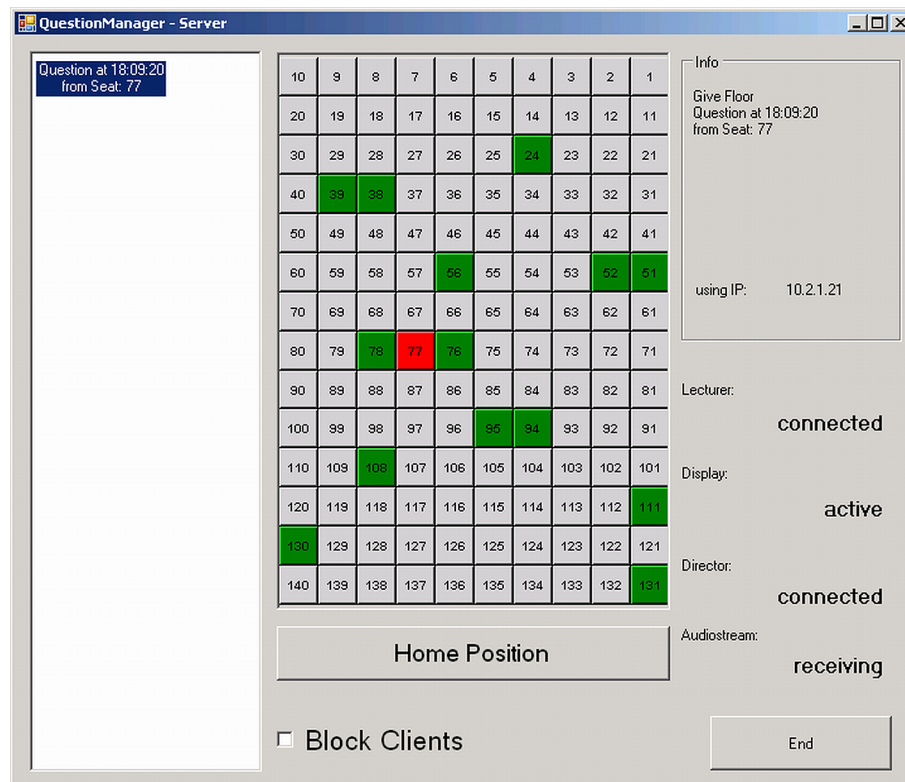


Figure 26: GUI of the QM server, client on seat 77 asking.

3.4. Sound Engineer

There is one crucial difference between the virtual sound engineer and all other modules of our system: the virtual sound engineer does not have a separate module but one part of it is implemented inside the Question Management software suite (see section 3.3.2.) and the other part is implemented inside the Audio-Video Mixer/Recorder module (see Section 3.5.).

Now, we focus on the tasks and the details of both parts of the virtual sound engineer.

3.4.1. Tasks to fulfill

The virtual sound engineer has to record all audio sources at a good quality level, send them to the AV Mixer/Recorder, and there these audio streams have to be mixed to the final audio track. As the pan-tilt-zoom (PTZ) cameras and one of the video servers not only support video streaming but also audio streaming, we utilize this feature. As shown in Figure 9 we use some analogue audio connections. One connection is used to feed the lecturer's voice into his or her presentation computer, in which all sounds of animations or simulations are mixed using the standard audio card and the mixing

application of the operating system. So, all the audio levels can easily be adjusted using the standard parameters.

The line-out socket is connected with the input of the video server of the slides. So, all audio signals and the associated visual content are encoded together and therefore automatically synced. Finally, one RTP stream for video, one RTP stream for audio and RTCP packets to sync both come out of the video server. The other analogue audio connection shown in Figure 9 joins the line-out socket of the computer running the virtual director and the QM server with the audio input of the PTZ audience camera with the built-in video server. The camera and its video server encode and sync the audio and video streams in the same way as the one used for the slides.

Utilizing these video servers for audio transportation eases the effort to bring all audio and video streams live to the AV Mixer/Recorder in a standardized way, which is a high-end machine located at our institute.

In the AV Mixer/Recorder software, all incoming streams are decoded. The second part of the virtual sound engineer now processes the three incoming audio streams. The first stream comes from the lecturer camera's video server but transports silence as we plugged the voice of the lecturer into his or her computer. The second stream comes from the slide's video server and contains the sampled sounds of the computer (for example, from an animation) and of the lecturer. The last data stream comes from the audience camera's video server and contains the sampled audio of the questioners, or silence when no-one is asking. The LongShot camera's video server does not transport audio data at all as it is not audio enabled.

All incoming audio data are processed with the same procedure by the virtual sound engineer in the AV Mixer/Recorder. At first, we apply a noise gate i.e., we analyze whether the audio data contains silence or sound; this makes sure that only those parts containing a valid signal get processed. This is necessary as silence in sampled audio data is useless. It does not stay on the negative infinity decibel (dB) mark of a waveform representation but waves around it at a really little volume level. If we processed it in the standard way we would get a huge disturbing noise during normalization of the audio volume level, which is the next step. Figure 27 shows this effect.

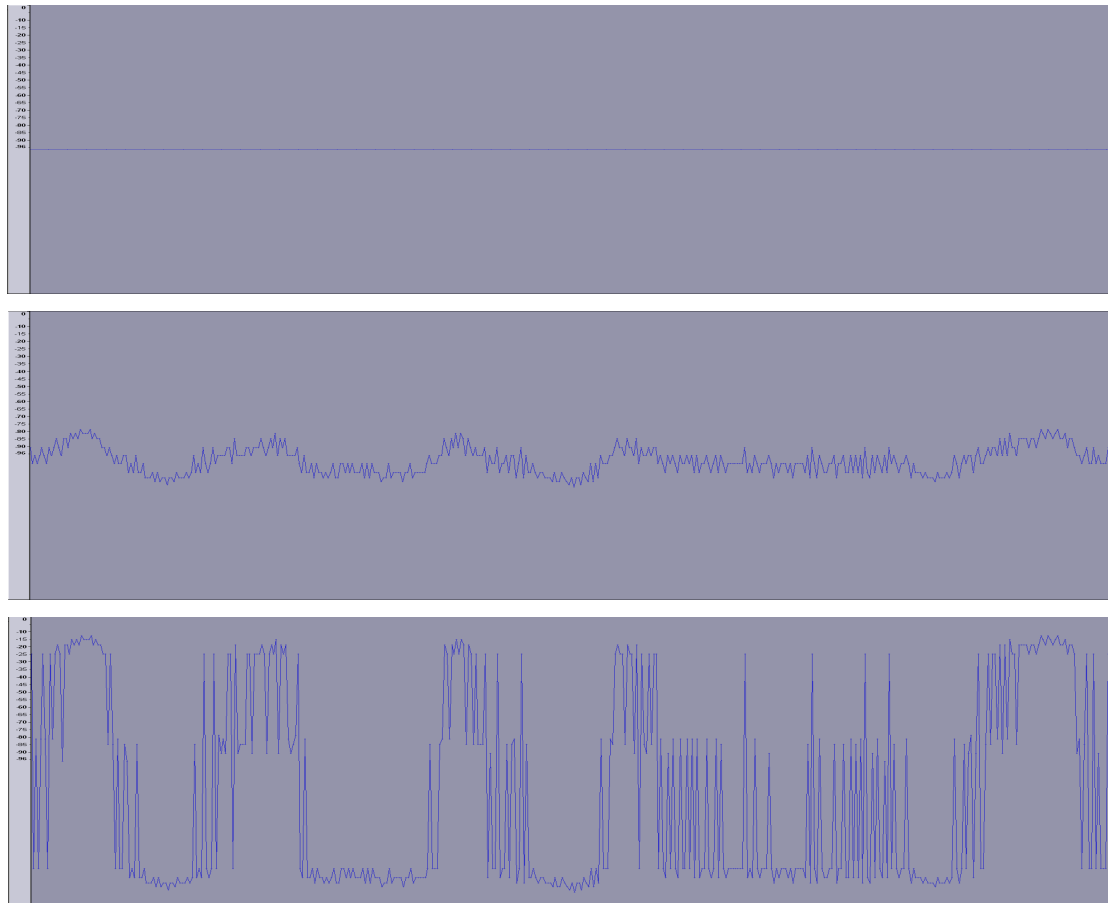


Figure 27: Ideal silence (top), minimal noise as silence (middle), and this minimal noise after normalization (bottom).

Normalizing the audio volume level is a process in which the maximum peak of sampled audio data and its distance factor to a given maximum is evaluated. Now, all audio data get amplified or dampened by this factor. It makes sure that no audio signal is louder than the given maximum, and different audio streams are adjusted to the same maximum volume.

Now, it is safe to mix all audio data streams together. Mixing of audio streams is done by summing up the values of all samples of the same point in time and then dividing up the sum by the number of audio streams. The result is the value of the mixed audio sample at this point in time.

The last step is to adapt the mixed audio data to the sampling rate, to the quantization resolution, and to the number of tracks needed for the output. A typical audio output signal used in conjunction with video is characterized by a sampling rate of 48 kHz, 16 bit, stereo, and a maximum volume level of -3 dB “*headroom*”.

3.4.2. Implementation Details

The sound engineer inside the QM software suite is split into two parts. One part is inside the QM questioner client where the sound is digitized into samples and send to the QM server where the samples are played back in order to feed the analog audio connection of the audience camera's video server.

When the questioner announces a question, the QM server and the QM questioner client negotiate a User Datagram Protocol (UDP) port for transmitting the audio samples. When the "*Questioner Acknowledged*" signal comes from the QM server, the questioner's client starts to sample the voice and to send the sampled data. Our client and server were inspired by (Konerow, 2007).

Due to the processing power of the PDAs, we decided a) to record the questioner's voice with one mono channel at a sample rate of 11.025 kHz and at a quantization of 16 bit and b) to use four buffers in a ring buffer setup with a buffer size of 10584 bytes, leading to a duration of 480 milliseconds per buffer. Tests have proven that our PDA is capable of switching the buffers every 480 ms without any jitter in the recorded audio, and of sending about 18 UDP packets via WLAN per second. So, we have to make the trade-off between no jittering audio and a delay of nearly half a second before getting the audio. A delay of 480 ms is clearly visible as the lips do not move in sync with the spoken audio and is unfortunately disturbing for the user but inevitable with the equipment used for our prototype.

If a UDP packet gets lost a break in the audio signal is the consequence. The duration of such a break can be calculated easily: The Maximum Transmission Unit (MTU) of wireless LAN (802.11) consists of 2312 bytes. We must subtract the length of the IP header which is between 20 bytes and 60 bytes depending on optional fields and the length of the UDP header which is eight bytes. Overall, we still have a payload of $2312[byte] - 20[byte] - 8[byte] = 2284[byte]$ in the best case and a payload of $2312[byte] - 60[byte] - 8[byte] = 2244[byte]$ in the worst case. So, if a UDP packet gets lost at a sample rate of 11.025 kHz and at a quantization of 16 bit (i.e., 2 bytes), we encounter an audio break of about 104 ms.

$$\frac{\frac{2284[\text{byte}]}{2[\text{bytePerSample}]}}{11025[\text{SamplesPerSecond}]} \approx 0.10358[\text{seconds}] \approx 104[\text{ms}]$$

Definition/Formula 7: Calculation of the audio break duration in case of a UDP packet loss.

Fortunately, such a break is not too long to miss the meaning of a sentence.

The receiver also uses a ring buffer with four segments, each at a size of 10584 bytes, and writes the incoming data onto the sound card in order to playback the audio. Thus, the sound is passed into the video server of the audience camera. When the QM server signals that the lecturer is starting to answer the volume of the playback is muted before stopping and resetting the audio connection on both devices. In this way switching noises are avoided.

The video servers we use encode the audio with a μ -law codec. This is a codec used to carry speech, was developed for analog telephone lines. Sound quality could be improved if other video servers with a different codec were used. The codec compresses the sound into audio data with a sampling rate of 8 kHz and 8 bits per sample before they are streamed over the network to the AV Mixer/Recorder.

Inside the AV Mixer/Recorder, the second part of the virtual sound engineer is implemented. The RTP audio streams are read and decoded so that we achieve standard audio pulse code modulation (PCM) samples, still at a sample rate of 8 kHz but already at a quantization of 16 bits. The μ -law codec uses a non-linear characteristic with 15 segments to map linear digital samples with a resolution of 12 bits to the 8 bits used for transmission. This is why after decoding the quantization is set to 16 bits which is a very common value. After having reconstructed the original 12 bit linear audio sample, the value gets converted to 16 bits by adding four zero bits at the least significant end.

Now, the maximum value of the current incoming decoded sample set is determined to make sure that this set of samples gets processed only if their maximum value is greater than 200. This threshold has been evaluated as useful for our setup; it is not exaggerated as the theoretical maximum value is 32767, and it represents a *noise gate* which suppresses all volumes below it. For example, a murmuring audience gets

mutated. The necessary amplifying factor for this set is determined including headroom of 3 dB. This is the final equation:

$$factor = \frac{32767}{SamplesetMaximumValue} * 0.705$$

Definition/Formula 8: Calculating the amplifying factor including 3 dB headroom.

The entire set is multiplied by this factor.

The next step is to mix all audio sources by summing up the values of all samples of the same point in time and then dividing the sum by the number of audio sources. The result is the value of the mixed audio sample at this point in time. Here is the corresponding source code:

```
public Int16[] Mix(Int16[] myPCM1, Int16[] myPCM2, Int16[] myPCM3)
{
    long minL = Math.Min(myPCM1.Length, myPCM2.Length);
    minL = Math.Min(minL, myPCM3.Length);

    Int16[] result = new Int16[minL];

    for (int count = 0; count < minL; count++)
    {
        result[count] = (Int16)Math.Round((double)(myPCM1[count] + myPCM2[count] +
            myPCM3[count]) / 3);
    }
    return result;
}
```

The next step is to resample the data from the incoming sampling rate of 8 kHz to the output sampling rate of 48 kHz. This means that we have to interpolate six outgoing samples for any of the incoming samples. The easiest way of interpolating data is a linear interpolation which is indeed sufficient. Figure 28 shows two times a part of a 440 Hz sine tone, sampled at 48 kHz (top) and sampled at 8 kHz (bottom). The small crosshairs along the curves represent the samples.

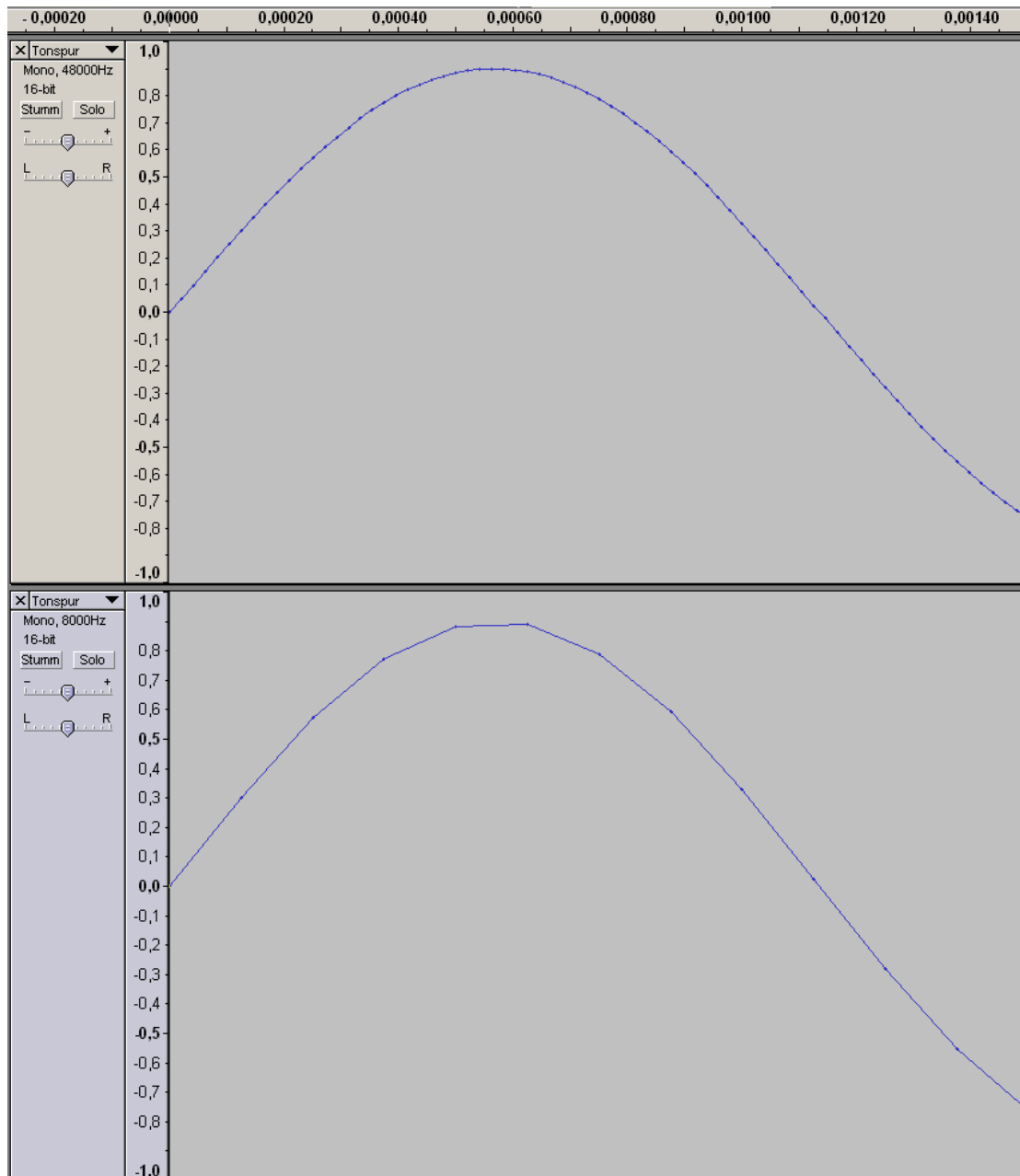


Figure 28: Part of a 440Hz sine tone at 48 kHz (top) and at 8 kHz (bottom).

Here is the source code which does the resampling:

```
public Int16[] Resample(Int16[] myPCM, int srcHz, int dstHz)
{
    int factor, x1, x2, resultIndex;
    Int16 y1, y2;
    Int16 newValue;
    double m, b;
    Int16[] result;

    //set basics once per call
    factor = (int)(dstHz / srcHz); //(48000 / 8000)
    x1 = 0;
    x2 = 1 * factor;

    //prepare arrays
    result = new Int16[factor * myPCM.Length]; //1channel, factorOfSampleRate(=6)
```

```

//run over whole sample set
for (int count = 0; count < myPCM.Length - 1; count++)
{
    //take the two next samples
    y1 = myPCM[count];
    y2 = myPCM[count + 1];

    //linear interpolation function parameters
    m = ((double)(y2 - y1) / (double)(x2 - x1));
    b = y2 - (m * x2);

    //do resampling 8kHz --> 48kHz
    for (int count2 = 0; count2 < factor; count2++)
    {
        //linear interpolation: f(x) = mx + b
        newValue = (Int16)Math.Round((m * count2) + b);

        //write into result array
        resultIndex = (count * factor) + count2;
        result[resultIndex] = newValue;
    }
}

return result;
}

```

At last, we double the channels to create pseudo stereo by using the mono data twice. We repeat the resampled data so that the original sample order “ABCD” results in “AABBCCDD”.

Now, we achieved the typical format of audio for video at a sample rate of 48 kHz, 16 bit quantization, two channels (stereo), and signal headroom of -3 dB.

3.5. Audio-Video Mixer/Recorder

The Audio-Video Mixer/Recorder is the last link in the chain. The two different streams the used cameras respectively the used video servers provide are a Motion-JPEG stream and an MPEG stream. While Motion-JPEG streams consist of separate JPEG images transmitted after each other, the MPEG-stream consists of subsequent Groups of Pictures (GOP). Inside one GOP there are one I-frame followed by P-frames, eventually amended by B-frames depending on the target of the stream. We use GOPs consisting of one I-frame followed by seven P-frames (IPPPPPPP). We need no B-frames as we do not want to do a reverse playback. The I-frame is an intra frame coded image similar to an JPEG image, the P-frames are so-called predictive images which contain the motion vectors of (sub-)pixels dependent onto the preceding I-frame and if existing also to preceding B-frames. B-frames are bi-directional predictive frames and are dependent onto the preceding and the succeeding I-frame. Thus, MPEG streams are better suited if the entire stream is going to be processed and/or displayed as they provide a better data compression rate than MJPEG streams. The latter are better suited if arbitrary single images should get extracted as they do not

have any temporal dependency. Therefore, the virtual cameraman uses either single JPEG images, separately requested, or the MJPEG-streams to easily extract single images, and the AV Mixer/Recorder uses the MPEG streams as it needs the continuous stream for recording.

Like the archetypes of AV Mixers in the broadcast world ours is built for live production: it is able to process the incoming audio and video streams in real time. This is correct for our prototype until it comes to recording the final audio and video track. Due to time constraints for this dissertation, we had to find a compromise. Instead of implementing new audio and video sources for the Windows Driver Model (WDM), which would be the best way for AV quality but would force us to implement in a programming language different from the language of our system. We tried to use the application programming interface (API) of QuicktimePro as it is described in (Cromie, 2006). Unfortunately, it turned out that it relies on the *Single-Threaded-Apartment* model; in contrast to the default setting of Microsoft's C# programming language. As the AV Mixer/Recorder has to handle four different incoming video streams and up to three incoming audio streams plus two outgoing streams for audio and video, it simply was not conceivable to leave our *Multi-Threaded-Apartment* model because of the many concurrent threads already used.

The compromise we finally chose was to write the fully processed images to disk as single files in JPEG-format while the final audio data were written into a standard WAVE file. After the end of the lecture, all files are automatically converted into the final DV-AVI file, containing the audio and video tracks. Our system is still live compatible as all the processing is done in real time; only the final encoding into the output format is done offline.

We also like to mention that the tasks the AV Mixer/Recorder has to fulfill in real time require a high-end workstation computer as it handles five uncompressed video streams and three uncompressed audio streams plus up to three video mixing streams and up to two audio mixing streams. The maximum is reached when two PiP-images are cross-faded and the final left and right channels need a different mixing. Overall, the maximum amount of data handled per second is enormous:

uncompressedVideo(RGB) =

$$\begin{aligned} & \text{FramePixelWidth} * \text{FramePixelHeight} * \text{BytesPerPixel} * \text{FramesPerSecond} = \\ & 720 \text{ [Pixel]} * 576 \text{ [Pixel]} * 3 \text{ [Byte/Pixel]} * 25 \text{ [Frames/Second]} = \\ & 31104000 \text{ [Byte/Second]} = 29.6630859375 \text{ [MB/Second]} \end{aligned}$$

uncompressedAudio(Mono) =

$$\begin{aligned} & \text{SampleRate} * \text{BytesPerSample} * \text{NumberOfChannels} = \\ & 48000 \text{ [Samples/Second]} * 2 \text{ [Byte/Sample]} * 1 \text{ [channel]} = \\ & 96000 \text{ [Byte/Second]} = 0.091552734375 \text{ [MB/Second]} \end{aligned}$$

AmountOfDataHandledPerSecond =

$$\begin{aligned} & (8 * \text{uncompressedVideo}(\text{RGB})) + (5 * \text{uncompressedAudio}(\text{Mono})) = \\ & (8 * 31104000 \text{ [Byte/Second]}) + (5 * 96000 \text{ [Byte/Second]}) = \\ & 248832000 + 480000 \text{ [Byte/Second]} = 249312000 \text{ [Byte/Second]} = \\ & 237.762451171875 \text{ [MB/Second]} \end{aligned}$$

The result is a total amount of data of almost 238 megabytes (MB) per second. The workstation we use is a Dual-Xeon Quad-Core at 2.66 GHz with 4 Gigabytes of RAM.

3.5.1. Tasks to Fulfill

The AV Mixer/Recorder has to decode all incoming audio and video streams. As the incoming frame rate of the four video sources is not guaranteed to be exactly 25 frames per second because of the design of the video servers themselves, we need to store the incoming frames in buffers. For each incoming stream, one buffer is used. So, we make sure that at any point in time a valid frame is available in the four buffers.

The decoded audio data is directly stored as raw wave data as the incoming audio data rate does not vary and no UDP packet losses have ever been detected.

To process the decoded AV data, the AV Mixer/Recorder receives the commands from the virtual director and acts accordingly. For our prototype, we implemented the following commands: switch between two video sources, fade between two video sources, and generate a new PiP video source out of two original video sources. As all audio sources get mixed constantly, no explicit commands are needed.

The next task is to produce the final output frame precisely every 40 ms in order to achieve a frame rate of 25 frames per second. Finally, the output frames have to be saved onto the disk, joined with the mixed audio data, and converted into a video file format that can be used for any further processing, e.g., converting the video into a web-streamable format or a format used for downloading from a web-page. We therefore choose the AVI file container, with audio and video tracks coded in the commonly used DV-format, well known by the DV consumer camcorders. This format is an intra-frame coded format, which eases to cut the video file at any frame position without re-coding. Its characteristics are a frame rate of 25 frames per second at a resolution of 720 by 576 pixels when used in PAL format, and two audio tracks used for a stereo sound reproduction at a sampling rate of 48 kHz and using a quantization of 16 bits.

In order to stay compatible with live productions, at least the recording of the final video frames and the audio data have to be done in real time. Any coding needed for the DV-AVI format can be done afterwards but should be done automatically to achieve a consistent state.

Not necessary for the actual processing of the AV data but very beneficial for a human supervisor is a comprehensive status display which gets refreshed regularly. Figure 29 shows a screen shot of our status display.

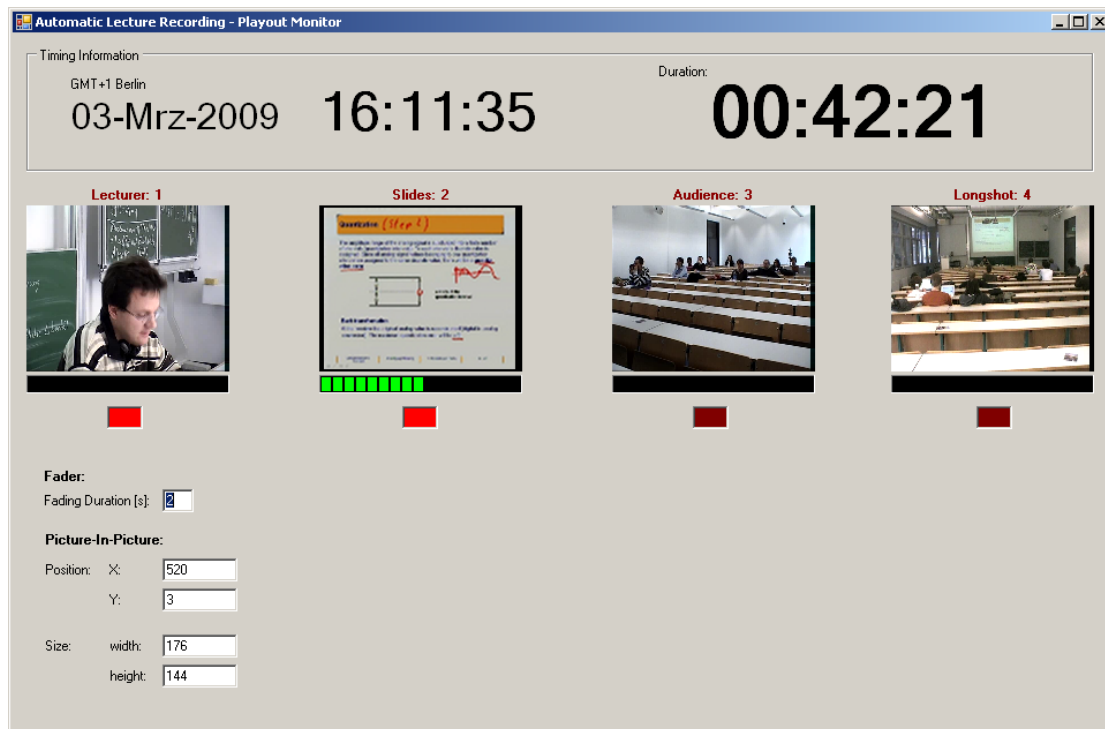


Figure 29: Screen shot of the status display of the AV Mixer/Recorder.

This display gets refreshed every second. It shows the current date and time in the upper left hand corner, as well as the duration of the current recording in the upper right hand corner. The main part of the display is used by four monitors showing the current image of all four video sources. Just beneath these monitors, the audio levels of the according audio sources are shown. For evaluation purposes, we recorded our lectures in parallel with Camtasia which is our standard procedure. Therefore we have to feed the speech of the lecturer into his or her computer on which the Camtasia software is running. Thus it is clear, that only the sound level under the monitor of the slides is visible instead of being shown separately beneath the lecturer and/or beneath the slides. Additionally, as no question is asked at the moment, the audience's audio level is muted due to our *noise-gate* implemented in the sound engineer. At last, no audio level is shown beneath the LongShot monitor as this source only provides a video stream.

In the next row under the audio level displays four *on air lights* are visible. In the Figure, the video sources of the lecturer and of the slides are currently on air in contrast to the other video sources. There are two possible reasons for two video sources being on air simultaneously:

a) a fade from one source to the other is just taking place, or

b) the current output shows both sources (the slides and the lecturer) picture-in-picture.

The rest of the status display contains the parameters used while mixing the video. It defines the duration of a cross-fade in seconds, and it defines the position and the size of the picture-in-picture image. The screen shot shows the default values used in our prototype; they can be changed at runtime if necessary.

3.5.2. Implementation Details

The AV Mixer/Recorder is a complex module and the central instance of our prototype. Figure 30 shows an overview on the structure of all data and AV streams processed.

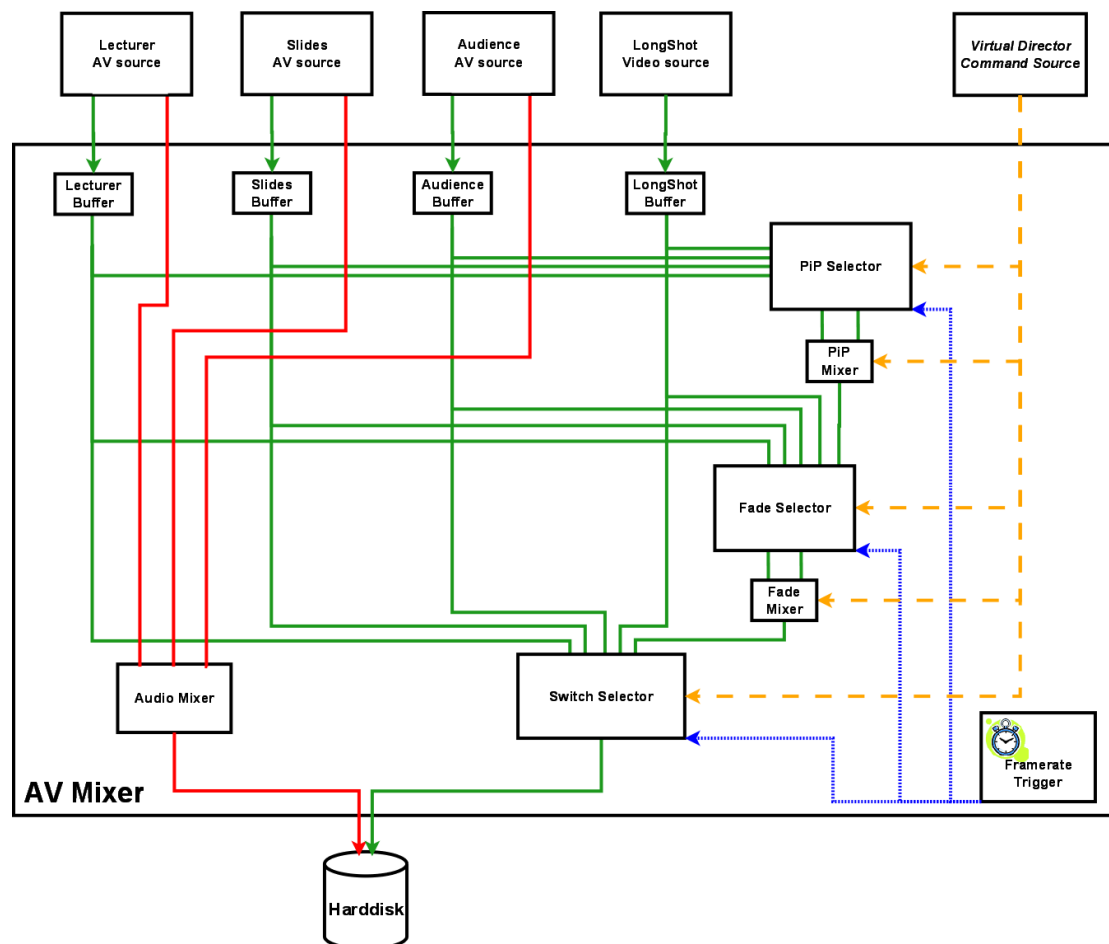


Figure 30: Overview of the AV Mixer/Recorder.

The red lines and arrows show the audio streams getting mixed and stored on the hard disks while the green lines and arrows show the video streams. As already mentioned, the frame rates of the four video sources are not fixed at 25 frames per second, and we

therefore introduced frame buffers for each input. Every time the incoming frame is complete, an event is thrown and fills the buffer with its frame. Shown by the dotted blue arrows, the *frame rate trigger* inside the AV Mixer/Recorder signals the reading of the currently needed buffers every 40 ms to create the current output image. Thus, we achieve a precise frame rate of 25 frames per second and reduce the access to the video buffers to the necessary minimum. This is important as the access to the buffers has to be controlled by *monitors* in order to synchronize the access of the threads writing data into it and reading data from it. The fewer the number of threads wanting to get access to a buffer the easier it is to synchronize them. The instructions of the virtual director shown as the yellow dashed arrows select which buffers are to be read, whether a PiP image has to be created, and whether a hard cut or a cross-fade should be used.

When the AV Mixer/Recorder is started it waits for the video servers to be accessible. It checks whether they react properly on an Internet Control Message Protocol (ICMP) ping. After the ping is replied correctly, it requests the AV streams by using the RTSP protocol via the Tao.FFMPEG interface, see the source code below.

```
public MediaFile(String URLfilename)
{
    int temp = URLfilename.LastIndexOf(":");
    string host = URLfilename.Substring(7,temp-7);
    while (!isPingable(host, out RTT))
    {
        Thread.Sleep(1000);
    }

    // Register protocols and codecs with FFMpeg
    FFMpeg.av_register_all();

    // Open stream with FFMpeg
    if (FFMpeg.av_open_input_file(out pFormatContext, URLfilename, IntPtr.Zero, 0,
        IntPtr.Zero)< 0)
        throw new Exception("Unable to open stream");

    // Get stream info
    if (FFMpeg.av_find_stream_info(pFormatContext) < 0)
        throw new Exception("Unable to find stream info");

    // Get context
    FFMpeg.AVFormatContext formatContext =
    PtrToStructure<FFMpeg.AVFormatContext>(pFormatContext);

    // Loop through streams in this file
    for (int i = 0; i < formatContext.nb_streams; ++i)
    {
        FFMpeg.AVStream stream = PtrToStructure<FFMpeg.AVStream>(formatContext.streams[i]);
        FFMpeg.AVCodecContext codecContext =
            PtrToStructure<FFMpeg.AVCodecContext>(stream.codec);

        // Get codec
        IntPtr pCodec = FFMpeg.avcodec_find_decoder(codecContext.codec_id);
        FFMpeg.AVCodec codec = PtrToStructure<FFMpeg.AVCodec>(pCodec);
        if (pCodec == IntPtr.Zero)
            continue;
    }
}
```

```

// Check codec type
switch (codecContext.codec_type)
{
    case FFmpeg.CodecType.CODEC_TYPE_AUDIO:
        // We only need 1 audio stream
        if (hasAudio)
            break;

        // Get stream information
        hasAudio = true;
        _samplerate = codecContext.sample_rate;
        _bitsPerSample = codecContext.bits_per_sample;
        audioStream = stream;
        _channels = codecContext.channels;
        originalAudioFormat = codecContext.sample_fmt;
        audioTimeBase = (double)codecContext.time_base.num /
            (double)codecContext.time_base.den;
        aFrameSize = 480;

        // Update codec context
        Marshal.StructureToPtr(codecContext, stream.codec, false);

        if (FFmpeg.avcodec_open(stream.codec, pCodec) < 0)
            throw new Exception("Unable to open audio codec");
        break;

    case FFmpeg.CodecType.CODEC_TYPE_VIDEO:
        // We only need 1 video stream
        if (hasVideo)
            break;

        // Get stream information
        hasVideo = true;
        width = codecContext.width;
        height = codecContext.height;
        videoStream = stream;
        originalVideoFormat = codecContext.pix_fmt;
        videoTimebase = (double)codecContext.time_base.num /
            (double)codecContext.time_base.den;

        // Update codec context
        Marshal.StructureToPtr(codecContext, stream.codec, false);

        if (FFmpeg.avcodec_open(stream.codec, pCodec) < 0)
            throw new Exception("Unable to open video codec");

        break;
}
}

//If no video found
if (!hasVideo)
    throw new Exception("No video codecs or streams found");

isPrepared = true;
}

```

For each AV source, a separate thread is started which receives and decodes the image and audio data. The source code of this routine is shown in Appendix 7.2.2. We extract the crucial part of throwing the corresponding events here:

```

#region video stream
// Is this a packet from the video stream?
if (packet.stream_index == videoStream.index)
{
    // Decode video frame
    int length = FFmpeg.avcodec_decode_video(videoStream.codec, vFrame, ref
        got_picture, packet.data, packet.size);

    // Did we get a video frame?
    if (got_picture != 0)

```

```

{
    LatestFrame = YUV2RGB(PtrToStructure<FFmpeg.AVFrame>(vFrame), 720, 576);
    //event senden
    this.myFrameReady.Invoke();
}
break;
}
}
#endregion

#region audio stream
if (packet.stream_index == audioStream.index)
{
    aFrame = new byte[aFrameSize];
    paFrame = Marshal.UnsafeAddrOfPinnedArrayElement(aFrame, 0);

    //Decode audio frame
    int length = FFmpeg.avcodec_decode_audio(audioStream.codec, paFrame, ref
        aFrameSize, packet.data, packet.size);

    //did we get an audio frame?
    if (length > 0)
    {
        this.AudioReceived(aFrame);
        aFrame = null;
        break;
    }
}
#endregion

```

The call `YUV2RGB(PtrToStructure<FFmpeg.AVFrame>(vFrame), 720, 576)` converts the AVFrame structure of FFMPEG which contains the image in YUV values with 4:2:2 chroma sub-sampling into a regular bitmap consisting of RGB values with 4:4:4 chroma sub-sampling. This is the format in which the bitmaps are written into the video buffers. Fortunately, FFMPEG is able to do the conversion, but before some preliminary steps have to be done:

```

private Bitmap YUV2RGB(FFmpeg.AVFrame yuv, int width, int height)
{
    //convert image from YUV to RGB
    IntPtr pYUV = FFmpeg.avcodec_alloc_frame();
    Marshal.StructureToPtr(yuv, pYUV, false);

    Bitmap bmp = new Bitmap(width, height, PixelFormat.Format24bppRgb);
    BitmapData bd = bmp.LockBits(new Rectangle(0, 0, bmp.Width, bmp.Height),
        ImageLockMode.WriteOnly, PixelFormat.Format24bppRgb);

    // Create RGB frame
    IntPtr rgbFrame neu = FFmpeg.avcodec_alloc_frame();
    FFmpeg.avpicture_fill(rgbFrame neu, bd.Scan0, (int)FFmpeg.PixelFormat.PIX_FMT_BGR24,
        width, height);

    // Convert video frame to RGB
    IntPtr ps11 = FFmpeg.sws_getContext(width, height,
        (int)FFmpeg.PixelFormat.PIX_FMT_YUV420P, width, height,
        (int)FFmpeg.PixelFormat.PIX_FMT_BGR24, 4, IntPtr.Zero, IntPtr.Zero,
        IntPtr.Zero);

    FFmpeg.AVFrame rFrame neu = PtrToStructure<FFmpeg.AVFrame>(rgbFrame neu);

    FFmpeg.sws_scale(ps11, pYUV, yuv.linesize, 0, height, rgbFrame neu,
        rFrame neu.linesize);

    bmp.UnlockBits(bd);

    // Free memory
    FFmpeg.av_free(rgbFrame neu);
    FFmpeg.av_free(pYUV);
    FFmpeg.av_free(ps11);

    return bmp;
}

```

}

The frame rate trigger cannot be realized by the standard timer component of Visual Basic .NET or C# .NET. This timer has an accuracy of about 100 ms even if it is possible to set the timer period in milliseconds. So, it was not possible to achieve a frame rate of 25 frames per second as this requires a period between two timer ticks of exactly 40 ms. Instead, we have used the Win32 multimedia timer functions of the *winmm.dll* which allows a precise resolution of up to 1 ms. Setting up this timer is easy:

```
_FrameOutTimer = new Multimedia.Timer();
_FrameOutTimer.Mode = Multimedia.TimerMode.Periodic;
_FrameOutTimer.Period = 40;
_FrameOutTimer.Resolution = 1;
_FrameOutTimer.Tick += new EventHandler(_FrameOutTimer_Tick);
```

As we have already described the audio processing in detail in the sound engineer section we focus on the video image processing here. The bitmaps read out of the buffers are handed over to the PiP mixing engine if necessary. The routine creating the PiP image takes six parameters: the background image, the PiP image, the x- and y-coordinates the PiP should be placed at, as well as new dimensions, i.e., the width and the height of the imposed image. This routine is shown here:

```
private Bitmap BmpPiP(Bitmap Background, Bitmap PiP, int x, int y, int w, int h)
{
    if (Background != null && PiP != null)
    {
        Bitmap d = new Bitmap(Background.Width, Background.Height);
        Graphics g = Graphics.FromImage(d);
        g.DrawImage(Background, 0, 0);
        g.DrawImage(PiP, new Rectangle(x, y, w, h), 0, 0, d.Width, d.Height, GraphicsUnit.Pixel);
        g.Dispose();
        Background.Dispose();
        PiP.Dispose();

        return d;
    }
    else
        return null;
}
```

It makes extensive use of the functionality of the graphics object and its commands, like DrawImage: it enables an easy and quick processing of bitmaps. DrawImage is a powerful command as it allows resizing and imposing images in a single call. But it can provide even more features. For our prototype, we still need a way to cross fade between two images. We therefore use a ColorMatrix to gradually change the transparency of an image. The ColorMatrix is a 5 by 5 transformation matrix, to be more precise, a homogeneous matrix. This type of matrices allows multiple transformations by a simple matrix multiplication, e.g., scaling, translation, etc. Two important homogeneous matrices for basic transformations are shown below:

$$\begin{aligned}
 \text{Translation_matrix} &= \begin{pmatrix} 1 & 0 & 0 & 0 & tr \\ 0 & 1 & 0 & 0 & tg \\ 0 & 0 & 1 & 0 & tb \\ 0 & 0 & 0 & 1 & t\alpha \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \\
 \text{Scaling_matrix} &= \begin{pmatrix} sr & 0 & 0 & 0 & 0 \\ 0 & sg & 0 & 0 & 0 \\ 0 & 0 & sb & 0 & 0 \\ 0 & 0 & 0 & s\alpha & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}
 \end{aligned}$$

Definition/Formula 9: Translation and Scaling by homogeneous matrices.

It is obvious that it is easy to combine translation and scaling into one matrix. For our application, the axis are not the typical x-, y-, and z-axis, but the red, green, blue, and alpha channels of the image. They use values of the type float. The float values on the main diagonal change the intensity of the color channel without any other manipulation, like shifting, etc. So, the intensity of the red color is determined by the value of element (0,0), the intensity of green by the value of element (1,1) and the intensity of blue by the value of element (2,2). At last, the transparency of the image is determined by the so called *alpha-channel*, its value is stored in the element (3,3). In our routine, shown below, we set up our color matrix in such a way that all color intensities are set to 100 percent, the w parameter for the homogeneous coordinates is set to 1, but the transparency of the alpha channel is set to the percentage we want to create. The ColorMatrix we use thus looks like this:

$$\text{ColorMatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & \text{percentage} & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

Definition/Formula 10: ColorMatrix for changing the transparency of an image.

Even calculating perspectives, orthogonal projection and rotations around an axis can be done easily by using such matrices. But coming back to our prototype, we only need to set the transparency which is the intensity of the alpha channel. After having this ColorMatrix, we apply it in the same call of the DrawImage function. Here is the according source code:

```

private Bitmap BmpBlend(Bitmap Source, Bitmap Dest, float percent)
{
    if (Source != null && Dest != null)
    {
        ColorMatrix cm = new ColorMatrix();
        cm.Matrix00 = cm.Matrix11 = cm.Matrix22 = cm.Matrix44 = 1f;
        cm.Matrix33 = percent;
        ImageAttributes ia = new ImageAttributes();
        ia.SetColorMatrix(cm, ColorMatrixFlag.Default, ColorAdjustType.Bitmap);

        Bitmap d = new Bitmap(Source.Width, Source.Height);
        Graphics g = Graphics.FromImage(d);
        g.DrawImage(Source, new Rectangle(0,0,Source.Width,Source.Height), 0, 0, d.Width,
            d.Height, GraphicsUnit.Pixel);
        g.DrawImage(Dest,new Rectangle(0,0,Dest.Width,Dest.Height), 0, 0, d.Width,
            d.Height,GraphicsUnit.Pixel,ia);
        g.Dispose();
        ia.Dispose();

        Source.Dispose();
        Dest.Dispose();

        return d;
    }
    else
        return null;
}

```

Finally, only the correct bitmap is chosen out of the four incoming buffers, out of the PiP-mixer, and out of the cross fade-mixer. In our prototype, we write this bitmap down onto the hard-disk.

In this chapter we described in detail the implementation of every part of the distributed system of the Automatic Lecture Recording prototype. We focused for every part on the according tasks to fulfill and on their realization. In the next chapter, we focus on the technical experience we have made.

4. Technical Experience

After implementing the different modules of our prototype of the Automatic Lecture Recording system we were interested in how they interact and collaborate. Therefore, we first did some simulation tests concerning the virtual director module. Then we extended our test to the interaction with the cameraman module as well as the interaction with the sensor tools module. Finally, we integrated the AV Mixer/Recorder into the system smoothly.

The hardware we used for our prototype consists on one hand of the computers for the lecturer, for the cameramen, the director, and for the AV mixing console. On the other hand it uses specialized hardware. In order to be precise, two PTZ cameras model 214 of the manufacturer AXIS for the lecturer and for the audience, two video servers with the model numbers 241 and 243s of the manufacturer AXIS for the overview camera and the slides. The slides are taken from the VGA interface of the lecturer's computer and converted into composite video signals beforehand, which get fed into the video server.

4.1. Experience with the director module

The virtual director as our core module contains the main idea how to simulate a real camera team with a distributed computer system. As explained above, we set up an extended FSM with contexts, transition possibilities, conditions, and some random factors but without any hard coded rules and without any fixed weights for the transitions. In order to prove that our system is worth the effort, we compared it with a simple FSM without contexts in which the next state is selected at random.

4.1.1. Evaluation of the Virtual Director

Although directing follows cinematographic rules, it is more an art than a science; there is always some range how to realize it precisely. Each human director has developed his or her own style of directing.

Generally, a director wants to show a new event as fast as possible to the spectators but under observation of cinematographic rules. This is due to the difference between motivated and unmotivated cuts: the first ones react on events of the environment while the latter ones simply occur to avoid unwanted durations of shots. In any case,

motivated cuts are more interesting for spectators as they provide new information in contrast to showing the same content at a different viewing angle.

We have defined criteria to measure our virtual director. The first measure is: “*After a new event has been signaled, how long does it take to show the corresponding shot?*” A typical example is: “How long does it take to show the slide after the lecturer started to annotate it?” At first sight, it apparently only answers the question: “How long does it take to show the requested shot?” But furthermore it is a measure of quality: if it takes too long to show the correct shot it is more likely that one will miss an important aspect or an action. If someone missed the question, it does not make much sense to listen to the answer. In general, a reaction to an event should be prompt; otherwise, the viewer may miss content.

In order to better evaluate the behavior of a finite state machine, we generalized and extrapolated this criterion asking: “*How long does it take to show **all** requested shots on the average?*” Because reacting quickly to an event assures two more aspects: First, the maximum duration of a shot gets naturally limited. As an example, just imagine a lecturer annotating the slides at first then explaining a detail and starting to gesticulate. The virtual director will show the slides as long as annotating produces significant motion rates and therefore events in the slides camera. As soon the lecturer stops annotating and starting gesticulating, the motion rate will be detected in the lecturer’s shot. This new event could be taken as the reason for switching quickly to the new camera if no other events supersede it. Second, a transition based on an event is always a motivated transition, a reaction on the environment, which satisfies our natural curiosity. As a result, the spectators will be less confused and more engaged.

While these first two criteria are fulfilled better if switching is done faster, the next criterion is fulfilled better if its percentage is less. It is the percentage of unmotivated transitions compared to all transitions. As unmotivated transitions may confuse the spectator more easily, it is better to have a smaller percentage.

The last criterion, in contrast, shows the percentage of shots fulfilled immediately i.e., in 0 seconds. This extraordinary situation occurs every time a requested shot is already on air, and this shot is the best way of providing the spectator with a continuous live production. Therefore, it is better to have a higher percentage of immediately ful-

filled shots. So, any of these measures not only concern a quick response but also the quality of a director module of the automated lecture recording system.

4.1.2. Testing Setup

We built the FSM of our approach and a second, simple FSM which used a random function to select the next state, only weighted by fixed values for each state, not reacting to any sensor input, for comparison. Let us call the first one “*sophisticated FSM*” and the second one “*simple FSM*”. As we did not have any sensor tools ready at this point in time, we created an application to manually record sensor inputs and the according timestamp for each sensor input during a lecture. We recorded some lectures and created a set of sensor inputs of a virtual “average” lecture. This average lecture has been sent to both finite state machines over and over again to simulate the run of multiple lectures including all sensor inputs, whether interpreted or not.

Even if it is always the same set of sensor inputs the result of the multiple runs are not identical as the duration of each shot is determined randomly by the *simple FSM* and it neglects any sensor inputs. The *sophisticated FSM* reacts on these sensor inputs but nevertheless the duration of each shot still varies. Therefore, it is very likely that at each run the FSM is in a different state at a specific point in time, and thus different possible transitions are available, and the effects of the sensor inputs vary as well. This leads to a similar but not identical behavior of the “sophisticated FSM” for each “replay” of the lecture. Table 9 shows an excerpt of the sensor inputs of our average lecture, including an acknowledged questioner at time-stamp “00:36:48:876”, a further inquiry at time-stamp “00:37:50:715”, and the final answer at timestamp “00:38:45:694”.

Table 9: Exemplary sensor inputs with timestamps of the “average lecture”.

LectureTime	Event-Text
00:35:49:120	Lecturer speaking
00:35:57:462	Slide annotation
00:35:59:855	Slide annotation
00:36:03:170	Slide annotation
00:36:05:603	Lecturer speaking
00:36:21:606	Slide annotation
00:36:23:419	Slide annotation
00:36:25:662	Audience inactive
00:36:27:635	Lecturer speaking
00:36:30:720	Audience inactive
00:36:32:112	Lecturer gesticulating

00:36:33:233	Lecturer moving
00:36:35:637	Lecturer gesticulating
00:36:36:949	Slide annotation
00:36:37:770	Lecturer speaking
00:36:38:381	Audience active
00:36:48:876	Questioner acknowledged
00:36:50:428	Questioner active
00:36:53:192	Lecturer active
00:36:54:434	Slide switch
00:37:02:065	Slide switch
00:37:06:601	Slide switch
00:37:12:710	Slide switch
00:37:27:551	Slide switch
00:37:29:154	Slide space
00:37:39:028	Lecturer speaking
00:37:40:330	Lecturer speaking
00:37:41:642	Lecturer gesticulating
00:37:43:434	Questioner active
00:37:50:715	Lecturer AnswerIncomplete
00:38:03:002	Questioner active
00:38:03:773	Questioner active
00:38:05:075	Questioner active
00:38:08:350	Questioner inactive
00:38:12:236	Slide switch
00:38:17:113	Slide switch
00:38:20:077	Slide switch
00:38:24:053	Slide annotation
00:38:24:293	Lecturer active
00:38:27:277	Lecturer speaking
00:38:27:768	Lecturer speaking
00:38:30:322	Lecturer speaking
00:38:36:200	Slide space
00:38:39:765	Slide space
00:38:42:209	Lecturer speaking
00:38:45:003	Questioner inactive
00:38:45:694	Lecturer AnswerOK
00:38:47:807	Lecturer active
00:38:52:413	Slide switch
00:38:55:838	Slide switch
00:38:58:232	Slide switch
00:39:02:197	Slide switch
00:39:04:821	Slide annotation
00:39:06:724	Slide space
00:39:07:856	Lecturer speaking
00:39:10:009	Audience inactive
00:39:15:116	Lecturer active
00:39:16:568	Lecturer speaking
00:39:32:080	Lecturer speaking

Very typical for questions are the multiple slide switch events after the question has been posed and after it was finally answered.

4.1.3. Simulation Results

Through our simulations we gained values of 4855 shots shown by the “*simple FSM*” and values of 5222 shots shown by the “*sophisticated FSM*” during the simulated lectures. The results of our test concerning the four different criteria defined above, which describe a virtual director’s behavior in detail, are presented now.

Our first criterion, the average duration to fulfill a requested shot, was done by the “*simple FSM*” in 4.85 seconds in contrast to the “*sophisticated FSM*” which did it in 3.56 seconds. Remembering that the faster an action or aspect is shown the easier it is to follow the recorded lecture, it comes out that the “*sophisticated FSM*” is better in dealing with this criterion.

From the average duration, we have a closer look on the minimum and on the maximum duration to fulfill a requested shot. The absolute minimum duration is zero seconds in case that the requested shot is already shown. This is true for 67.72 percent of shots shown by the “*simple FSM*”, but it is true for 71.49 percent of shots shown by the “*sophisticated FSM*”.

The statistical maximum duration to surely fulfill a requested shot is the maximum duration observed throughout the entire session. The “*sophisticated FSM*” reaches this goal after 193 seconds which is nearly 90 seconds faster than the “*simple FSM*” reaching it after 286 seconds. So, again for these two criteria the “*sophisticated FSM*” performs faster than the “*simple FSM*”.

Looking at the percentage of unmotivated cuts, which are cuts not requested by a sensor input but forced by the expired duration of the preceding shot, their values are in the same range. To be more precise, it is 15.03% for the “*simple FSM*” and 12.52% for the “*sophisticated FSM*” which is a slight advantage for the latter. An overview over these values is given in Table 10.

Table 10: Simulation results of both finite state machines.

	Simple FSM	Sophisticated FSM
Number of shots	4855	5222
Percentage fulfilled after 0 seconds	67.72%	71.49%
Average duration to fulfill 100% of the requested shots	286 sec	193 sec
Average duration to fulfill a requested shot	4.85 sec	3.56 sec
Percentage of unmotivated cuts	15.03%	12.52%

To give an impression of the characteristics of both finite state machines concerning the percentage of all the fulfilled requested shots over time, Figure 31 shows the values of the “*simple FSM*” as triangles and the values of the “*sophisticated FSM*” as darker printed diamonds. As assumed from the faster average and the shorter duration to reach 100%, the graph of the “*sophisticated FSM*” has a steeper curve and therefore fulfills more requested shots in less time.

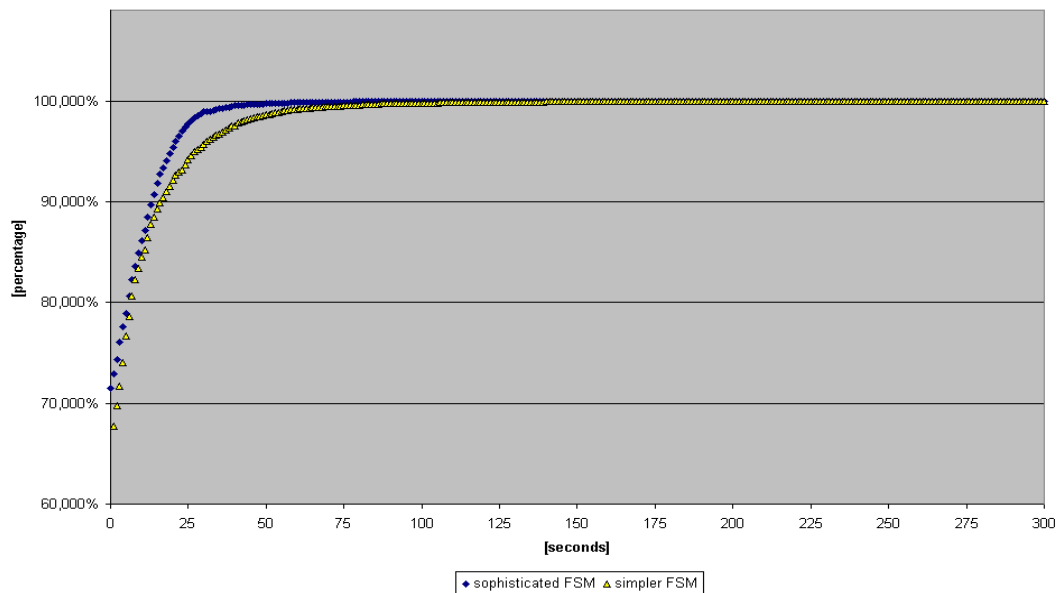


Figure 31: Percentage of fulfilled requested shots after n seconds.

The principle of the “*simple FSM*” has already proven its ability to act more or less satisfyingly as a video producing director in some implementations, e.g., in (Rui *et al.*, 2001), even though it tends to be predictive and uniform; but our approach, the “*sophisticated FSM*”, has shown that it is able to act much faster than the “*simple FSM*” and is thereby able to diversify the following states more easily based on the sensor inputs.

4.1.4. Overall Performance

Throughout the years of implementing, we always checked how the virtual director module decides and behaves. At first, it module produced only a log file, showing the active state number and the final probabilities of the selection process. Below there is a short snippet of such a log file:

```
...
ActivStateNo: 4
Possibility# 0, p=0.81 for NewStateNo: 3
Possibility# 1, p=0.81 for NewStateNo: 3
```

```

Possibility# 2, p=0.81 for NewStateNo: 3
Possibility# 3, p=0.855 for NewStateNo: 6
Possibility# 4, p=0.855 for NewStateNo: 6
Possibility# 5, p=0.7695 for NewStateNo: 4
Possibility# 6, p=0.7695 for NewStateNo: 4
Possibility# 7, p=0.81 for NewStateNo: 2
Possibility# 8, p=0.81 for NewStateNo: 2
Possibility# 9, p=0.15 for NewStateNo: 15
Possibility# 10, p=0.135 for NewStateNo: 7
ActivStateNo: 6
Possibility# 0, p=0.81 for NewStateNo: 3
Possibility# 1, p=0.81 for NewStateNo: 3
Possibility# 2, p=0.6885 for NewStateNo: 4
Possibility# 3, p=0.6885 for NewStateNo: 4
Possibility# 4, p=0.81 for NewStateNo: 5
Possibility# 5, p=0.81 for NewStateNo: 5
Possibility# 6, p=0.81 for NewStateNo: 2
Possibility# 7, p=0.81 for NewStateNo: 2
Possibility# 8, p=0.15 for NewStateNo: 15
Possibility# 9, p=0.135 for NewStateNo: 7
ActivStateNo: 5
Possibility# 0, p=0.81 for NewStateNo: 3
Possibility# 1, p=0.81 for NewStateNo: 3
Possibility# 2, p=0.81 for NewStateNo: 6
Possibility# 3, p=0.81 for NewStateNo: 6
Possibility# 4, p=0.81 for NewStateNo: 5
Possibility# 5, p=0.81 for NewStateNo: 5
Possibility# 6, p=0.81 for NewStateNo: 2
Possibility# 7, p=0.81 for NewStateNo: 2
Possibility# 8, p=0.15 for NewStateNo: 15
Possibility# 9, p=0.135 for NewStateNo: 7
ActivStateNo: 2
Possibility# 0, p=0.9 for NewStateNo: 3
Possibility# 1, p=0.9 for NewStateNo: 3
Possibility# 2, p=0.9 for NewStateNo: 2
Possibility# 3, p=0.81 for NewStateNo: 5
Possibility# 4, p=0.81 for NewStateNo: 5
Possibility# 5, p=0.7695 for NewStateNo: 5
Possibility# 6, p=0.855 for NewStateNo: 6
Possibility# 7, p=0.855 for NewStateNo: 6
Possibility# 8, p=0.15 for NewStateNo: 15
Possibility# 9, p=0.135 for NewStateNo: 7
ActivStateNo: 2
Possibility# 0, p=0.9 for NewStateNo: 3
Possibility# 1, p=0.9 for NewStateNo: 3
Possibility# 2, p=0.7695 for NewStateNo: 2
Possibility# 3, p=0.855 for NewStateNo: 5
Possibility# 4, p=0.855 for NewStateNo: 5
Possibility# 5, p=0.81 for NewStateNo: 5
Possibility# 6, p=0.9 for NewStateNo: 6
Possibility# 7, p=0.9 for NewStateNo: 6
Possibility# 8, p=0.15 for NewStateNo: 15
Possibility# 9, p=0.135 for NewStateNo: 7
ActivStateNo: 3
...

```

State numbers 2, 3, 4, 5, and 6 belong to the standard lecture context while state number 7 belongs to the question context and is only selected if a questioner gets acknowledged. Finally, state number 15 is the EndOfLecture state of the FSM, it is only selected when the lecture is over.

The total number of possibilities coming out of the active state is not limited, and is defined by the given FSM and all the possible ways of transition to another shot, like hard cut or cross-fade, for example. This is also the reason why more than one possibility can be used to access a new state. They only differ in the way they perform the transition, either as a hard cut or as a cross-fade.

In the next step, we have recorded the videos of all cameras and the audio of the slides camera. In addition, the director module wrote Edit Decision Lists (EDLs) in the style of the common but proprietary CMX3600 format, originally used for the machines of CMX Editing Systems which focused on on- and off-line editing in post-production in the 1970s and 1980s. The scheme of an EDL file is shown in Figure 32.

```

TITLE: Sequenz 01
000 AX V C 00:00:00:00 00:00:02:01 00:00:00:00 00:00:02:01
REEL AX IS CLIP 03-06-2008_15-31-29-LONGSHOT.mp4.avi

```

Figure 32: Explanation of an EDL entry.

The first line contains the title which is used to name the production. The second line shows the first real instruction for the resulting production. Its consecutive number “000” is followed by a token to identify the so-called “reel”. Originally, a reel is a film spool which e.g., contains a continuous shot of a scene. All reels together are called “footage”. Today, a reel is also the electronic representation of a reel called clip. The token gets mapped to the real part of the footage in the next line, containing comments. Here, the token “AX” gets mapped to the clip “03-06-2008_15-31-29-LONGSHOT.mp4.avi”. There are two ways to map tokens to clips: the first one is to define one token per clip by a comment line and only use different tokens per instruction. The second way is to always use the same token and every time map a different clip to that token. The latter way is used by Adobe Premiere Pro when exporting EDLs out of its time line. We adopted this way for our prototype as we wanted to use Premiere to finalize our production at that time.

The letter or letters in the orange-colored ellipse mark which tracks are involved by this instruction. Typical versions are: “V” – only the video track, “A” – only the mono audio track, “AA” – only the stereo audio tracks”, “A/V” – mono audio track and video track, and “AA/V” – stereo audio tracks and video track.

The letter in the cyan-colored ellipse describes the way of transitioning between shots. “C” stands for “hard Cut” and “D” for “Dissolve”.

The first two SMPTE timestamps set the “Play In” and “Play Out” points of the source reel, the last two SMPTE timestamps set the “Record In” and “Record Out” points of the resulting production.

This format can be easily imported by the editing software Adobe Premiere Pro. It accesses the recorded materials of the cameras, called “footage reels”. Below, we show an example of what an EDL of our prototype looks like:

```
TITLE: Sequenz 01
000 AX      V      C      00:00:00:00 00:00:02:01 00:00:00:00 00:00:02:01
REEL AX IS CLIP 03-06-2008_15-31-29-LONGSHOT.mp4.avi

001 AX      V      C      00:00:02:01 00:00:12:03 00:00:02:01 00:00:12:03
REEL AX IS CLIP 03-06-2008_15-31-29-LECTURER.mp4.avi

002 AX      V      C      00:00:12:03 00:01:03:11 00:00:12:03 00:01:03:11
REEL AX IS CLIP 03-06-2008_15-31-29-SLIDES.mp4.avi

003 AX      V      C      00:01:03:11 00:01:35:07 00:01:03:11 00:01:35:07
REEL AX IS CLIP 03-06-2008_15-31-29-LECTURER.mp4.avi

004 AX      V      C      00:01:35:07 00:02:01:20 00:01:35:07 00:02:01:20
REEL AX IS CLIP 03-06-2008_15-31-29-AUDIENCE.mp4.avi

005 AX      V      C      00:02:01:20 00:03:52:19 00:02:01:20 00:03:52:19
REEL AX IS CLIP 03-06-2008_15-31-29-SLIDES.mp4.avi

006 AX      V      C      00:03:52:19 00:06:03:16 00:03:52:19 00:06:03:16
REEL AX IS CLIP 03-06-2008_15-31-29-LECTURER.mp4.avi

007 AX      V      C      00:06:03:16 00:06:56:09 00:06:03:16 00:06:56:09
REEL AX IS CLIP 03-06-2008_15-31-29-LONGSHOT.mp4.avi

...

287 AX      V      C      01:24:58:01 01:25:08:13 01:24:58:01 01:25:08:13
REEL AX IS CLIP 03-06-2008_15-31-29-LONGSHOT.mp4.avi

288 AX      A      C      00:00:00:00 01:25:08:13 00:00:00:00 01:25:08:13
REEL AX IS CLIP 03-06-2008_15-31-29-SLIDES.wav
```

At this point in development, we recorded four camera video tracks and one audio track in parallel, and so the SMPTE timestamps of “Play In” and “Record In” as well as the SMPTE timestamps of “Play Out” and “Record Out” are always identical in a line. The last instruction of our EDL handles the audio tracks of the production. As at that time we only recorded one audio track, we use it for the entire production.

When all files are available to Premiere it produces the final video based on the director’s decisions given through the EDL. This process is called “on-lining” in broadcaster’s slang. Using EDLs gave us the first possibility to produce a real video based on the director’s decisions at a time when the AV Mixer/Recorder was not yet ready. An example of such a video was presented in (Lampi, Kopf & Effelsberg, 2008).

As soon as the AV Mixer/Recorder was functional, we automatically received the final result, as we had planned. During these steps of development, the diversity of the selected shots and transitions behaved as expected because the reaction on sensor inputs worked properly. We only had to make sure that too many similar sensor inputs did not lead to some kind of a “Denial-of-Service (DoS) attack”. Therefore, we intro-

duced a clearing instance purging all future inputs that were still in the message queue, leading to the same context as the one already active.

In order to keep the change of the context in the way it was planned, we also had to make sure that the FSM would not be stuck in the question or answer context forever in case a user forgot to click on the right button in the heat of the moment. For example, a lecturer may forget to hit the “Now Answering” button before starting to answer, or neither the lecturer nor the questioner closes the answer by clicking on the “Answer OK” button. So, we introduced time-outs for the question context and the answer context which are refreshed by any click on a relevant button of the question manager. Now, it is assured that neither the question context nor the answer context are left too early, and both contexts will finally be left to reset the active context to the standard lecture context.

Our experience with the system shows that for recording lectures our prototype is well suited. For adapting it to other contexts, the FSM and maybe some weights of the different shots or the values manipulating the possibilities during the transition selection need to be revised.

4.2. Experience with the cameraman module

The virtual cameraman’s main challenge is its real-time capability. There are many robust and stable image processing algorithms for numerous different tasks inside the MoCa library (MoCa, 2006) but many of them are used in an offline context as they process images loaded from disk and can take all the time they need.

For our prototype of the distributed Automatic Lecture Recording system, we need algorithms working fast enough for real-time, even if the result of an algorithm is not 100 percent perfect. This is true for the algorithms of image processing, for the algorithms controlling the camera, and for the control loop implementation of the virtual cameraman. We examine the virtual cameraman’s behavior under these criteria before summarizing the overall performance.

4.2.1. Performance of the image processing algorithms

The virtual cameraman’s main image processing algorithms are the motion detection and the skin color detection. On the one hand, they provide the measured values for

the cameraman's sensor input while, on the other hand, they trigger the controlling algorithms in order to react in an appropriate way to the occurrences in the images.

For motion detection, we have implemented two slightly different algorithms because of the different origin of the images. One algorithm is used only for the slides video server output which is normally characterized by a very static image. Only some converter noise has to be filtered out in order to avoid false alarms of detected motion. The other algorithm has to cope with images of the real world which, besides camera sensor noise, may contain arbitrary motion. It is therefore more complex to differentiate between motion in the foreground, which normally is the motion we want to detect, and motion in the background, like trees waving in the wind which is of no interest for us.

For the first algorithm, we use the *Frame Differencing* approach and simply determine the distance of two pixels in the RGB color space in the difference of two consecutive images to detect changes. In order to distinguish between converter noise and motion in the image, the distance must be larger than a threshold, calibrated for the video server.

As a trade-off between precision and speed, we decided to tile the image before applying any algorithm to it. Thus, we do not check the whole image on a per pixel base for motion but check whether the motion in a tile is larger than the threshold. The total percentage of motion in the image is then calculated by dividing the number of tiles in which motion was above the threshold by the total number of tiles in the image. This of course is only an approximate result but is still good enough for our purpose. The main advantage comes not only from a single algorithm but from combining it with others. For example, we will search for motion only in tiles which were already marked by the skin color detection algorithm when looking for a person moving around.

The latter algorithm to detect motion in real world images is a bit more complex. We use the *Background Subtraction* approach to avoid a background leading to false alarms. At first, we establish a background model for the image based on the *Running Gaussian Average* algorithm which gets initialized with the first image $BG_0 = IMAGE_0$ and kept up to date using Formula 11:

$$BG_i = (1 - \alpha) * BG_{i-1} + \alpha * IMAGE_i$$

Definition/Formula 11: Running Gaussian Average formula to update the background model.

The factor α defines how fast a new object gets integrated into the background model. It can take values in the range $[0;1]$. The closer the α value is to one, the faster new objects get incorporated into the background model. While an α value of one leads to the same behavior the *Frame Differencing* approach, so called “ghosts” will occur when using a smaller α value. The ghosts occur when, e.g., a slowly moving object gets incorporated into the background model before moving further. Again, it is a trade-off to choose this α value. For our prototype, $\alpha = 0.5$ works fine. We now subtract the background model from the current image and again determine the tiles in which motion occurred; in this way we are able to roughly determine the percentage of motion in the current image.

Both algorithms are very resource-friendly and run in real-time. So, their performances are definitely sufficient for our system.

For skin color detection, we only use one algorithm as we expect skin color only in real-world images. It is based on the algorithm of the MoCA library (MoCA, 2006) and consists of two steps: At first, the red values and the green values of an image get normalized in order to make the algorithm more robust against changes of the brightness, Formula 12 shows the details:

$$RED_{norm} = \frac{RED}{RED + GREEN + BLUE + 1}$$

$$GREEN_{norm} = \frac{GREEN}{RED + GREEN + BLUE + 1}$$

Definition/Formula 12: Normalizing red and green values for skin color detection.

The pixels are assumed to show skin color if both of their values are in the following ranges: $[0.37 \leq RED \leq 0.58]$ and $[0.26 \leq GREEN \leq 0.36]$.

This algorithm works fast and fairly well but cannot distinguish between real skin and items having a similar color. Therefore, we combined the tiles in which skin color was detected with the tiles in which motion was detected as most people are always moving a little bit. The result is sufficient for our needs and it still runs in real time.

Concerning the image processing algorithms, the virtual cameraman's performance definitely fulfills our needs. They are able to provide the necessary information for calculating the sensor inputs for the virtual director and for triggering the autonomous camera control procedures in real-time.

4.2.2. Performance of the camera controlling algorithms

Besides the parameters already mentioned in Section 3.2, the virtual cameraman module is also able to steer the pan as well as the tilt and zoom of the PTZ cameras. We defined a Cartesian coordinate system for our lecture hall and set the zero degree angles of the cameras parallel to the x-axis; in this way we made sure that there were no constraints concerning the valid ranges of pan and tilt angles of the cameras. For a precise comparison, we measured all lengths and positions using a laser-based distance measuring device. We allow two addressing modes for the cameras, absolute and relative; While the absolute addressing is used, e.g., for pointing the camera on a questioner, the relative addressing is used, e.g., for following a moving person. The built-in definition of the cameras of the manufacturer sets that negative values stand for angles left of the zero degree adjustment for panning and for angles below the horizontal adjustment for tilting. In order to keep the calculation of camera motion angles easy, we made sure that the cameras were located on the opposite site of the origin of the coordinate system, which means that negative angle values have the same meaning as those built into the cameras.

The coordinates of questioners in the room, as transmitted by the sensor tools module, refer to this coordinates system which enables us to use absolute addressing for all camera movements. First, we determine the distance vector between the position of the camera and the position of the questioner. The arc tangent of the x-value and the y-value of the distance vector result in the pan angle for the camera. If the target is left of the camera position the negative angle has to be taken. Second, we calculate the tilt angle in the same way; if the target is below the camera position, the negative angle has to be taken. Third, we want to show approximately three seats, the questioner and his or her right and left neighbors, to overcome possible position estimation errors of the indoor positioning system used in the sensor module. We defined 1.65 meters as the width of three neighbored seats (W_{orig}). We need three technical specifications of the camera: the maximum zoom factor (Z_{max}), the width of the optical sensor (W_{CCD}),

and the minimal focal distance (f_{\min}). Having the length of the three-dimensional distance vector (d), we first calculate the focal distance (f_{Dist}) needed to show an object of width ($W_{\text{orig}}=1.65\text{m}$) in this distance to fill the width of the image by means of the theorem on intersecting lines. Then, we calculate the necessary zoom factor (Z) by taking the ratio of f_{Dist} to f_{\min} . Of course, Z must be greater than or equal to 1, and Z must be less than or equal to Z_{\max} :

$$\begin{aligned}\frac{f_{\text{Dist}}}{W_{\text{CCD}}} &= \frac{d}{W_{\text{orig}}} \\ \Rightarrow f_{\text{Dist}} &= W_{\text{CCD}} * \frac{d}{W_{\text{orig}}} \\ Z &= \frac{f_{\text{Dist}}}{f_{\min}} \mid (1 \leq Z \leq Z_{\max})\end{aligned}$$

Definition/Formula 13: Calculating the zoom factor to frame a questioner.

As an example for a distance of four meters, the zoom factor results in:

$$Z = \frac{(W_{\text{CCD}} * \frac{d}{W_{\text{orig}}})}{f_{\min}} = \frac{(0.0036 * \frac{4}{1.65})}{0.0041} \approx 2.13$$

Definition/Formula 14: Calculating the zoom factor for a distance of four m.

The last step is to map the calculated zoom factor to a zoom parameter value of the camera. As no formula was available to do that, we used splines to approximate the values. The web interface of the cameras of our manufacturer allows setting an integer zoom factor and then reading which parameter value was used. Having the values for all integer zoom factors from 1 to 18, we used these data pairs to calculate the cubic splines. Thus, we are able to precisely calculate the zoom parameter value by using the correct spline of the according zoom factor. Now, the pan and tilt angles and the zoom parameter value are sent to the camera interface.

As all calculation steps needed for the absolute addressing of the cameras only use basic arithmetic operations, raising to the power of at most three and applying the arc tangent, the entire calculation can easily be done in real-time.

The second way to control the cameras is relative addressing. It is useful for a camera follow-up of a person. At first, the number of faces in the image is determined. If no face is found nothing happens. If one face is detected, it is used. If more than two

faces are found, the group of faces that takes the largest space in the image is used; if exactly two faces are found, some complex checks take place:

- Check whether one face is above the other, take the upper one.
- Check whether both faces are close together, and then treat them as one area.
- Check the designated alignment. Take the left face if left alignment is desired or take the right face if right alignment is desired.

After these checks, one face area remains. The coordinates of its center are determined and compared with the coordinates of the alignment point, either more on the left side or more on the right side. If the difference is above a threshold, the center coordinates of the face area are set as the new center coordinates of the image, and the values are sent to the camera interface.

The calculation for the new center is very fast but there is a disadvantage of the cameras we use which is more severe for relative addressing than for absolute addressing: the cameras do not report when they have finished an operation. While absolute addressing sets the new coordinates once, there is no need to wait for a completion acknowledgment from the camera. In contrast, the relative addressing is used for follow-ups of the camera, and therefore it is an enduring process. As it relies on image processing, it depends on an image taken after the last movement is finished. The time span between two consecutive useful images is about 1.5 seconds. Therefore, it is impossible to follow-up fast moving persons or persons who are very close to the camera, as even small position changes from one image to the other lead to large changes of the camera angles.

A human cameraman overcomes this problem by first zooming out and only if this measure is not sufficient he or she follows the person. That is why we implemented such an algorithm taking the motion of the image into account. Every time the virtual cameraman module detects motion, it repeatedly zooms out a little until the percentage of motion in the image is below a threshold. If there is only little motion in the image for a while, the virtual cameraman again zooms in. The advantage is that zooming is performed very quickly by the cameras so we do not have to wait until it is finished. In addition, we have made sure that this algorithm is executed before we try to follow-up a person.

Finally, the algorithms themselves are definitely fast enough, and in most combinations they fulfill our expectations. Especially, all the algorithms not relying on a finished camera operation are working perfectly. Only the follow-up of a person is a little slow but it is still sufficient for the lecturer sitting in front of the audience, as is always the case in our scenarios. Nevertheless, it should be possible in future work to optimize the behavior of the virtual cameraman module in this respect, for example by fostering parallelization of some algorithms.

4.2.3. Overall Performance

The virtual cameraman module has proved its ability to process all necessary tasks in real-time. The control loop approach works as expected and provides all necessary steps. These steps, the algorithms of image processing and controlling the camera accordingly as well as the communication with the virtual director, have a certain amount of complexity which should not be underestimated. That is why we put the cameraman to sleep for 550 ms in each run of the loop. This value is configurable, it has been evaluated to work well for the computer we elected to run all four cameramen on. Figure 33 shows an example of the status message displays of three of the four cameramen during a lecture recording as the fourth did not properly fit in the image any more.

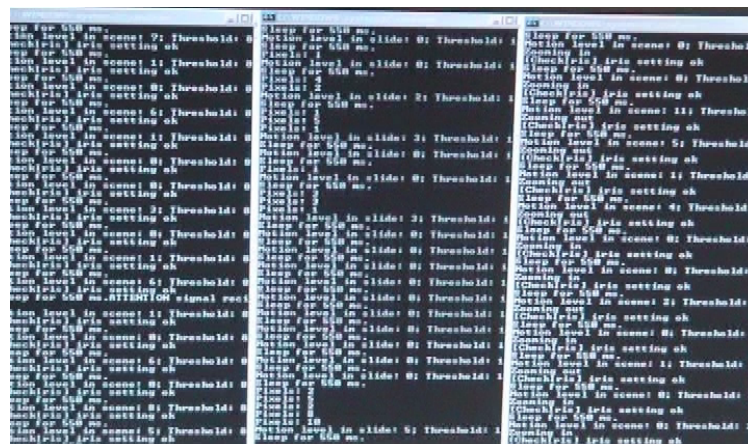


Figure 33: Exemplary status message displays of three virtual cameramen.

The complex combination of image processing algorithms and under some circumstances the waiting for the camera to finish the last movement order can be improved in future work by introducing asynchronous ways of calculating new instructions, sending them, getting feedback of instruction completion, and minimizing sleeping

times versus the load generated. Nevertheless, the virtual cameraman module is still sufficient for the activities in a lecture hall as we made sure that all instructions from the director get processed even if they are transmitted while the cameraman is sleeping.

4.3. Experience with the Sensor Tools module

While we have handled each part of the Sensor Tools module separately in the previous sections, we now will focus on our experience of the collaboration of all parts of the module. In the following sections, we examine the question – answer interaction control, the position estimation and event reporting, and the audio streaming.

4.3.1. Experience with the Question – Answer Interaction Control

The experience with the question – answer interaction control consists of three different aspects: its applicability to the real world, its acceptance by the students, and its workload for the lecturer.

Applicability to the real world

We started with the most obvious and simple approach to map the question – answer interaction to our implementation by using a so-called “paired approach”. This means that with any interaction the active part toggles between the questioner and the lecturer. For example, the standard way is described as: “Questioner announces” – “Lecturer gives the floor”, “Questioner asks” – “Lecturer answers”, “Lecturer checks answer against expectations” – “Answer acknowledged”. Even for further inquiries, this paired approach works if in the last step the “Answer is not acknowledged” is interpreted as a new request. Then the interaction continues like this: “Answer not acknowledged (Questioner announces)” – “Lecturer gives the floor” and so on. Even deferring questions to a later time and denying or stopping questions are easily mapped using this paired approach.

During our tests, we found out that the lecturer and also the students coped very well with this approach. Only for further inquiries, a small hint was necessary, explaining how to initiate it. Nevertheless, this approach is limited to one questioner at a time. In case of comments of other participants or even a discussion among multiple participants, this leads to get the paired approach overcharged; those types of interactions should be implemented in future work.

Acceptance by the students

From the very first time of testing the Question Management module in the lecture, there was a spontaneous positive reaction of the students. The most interesting change in contrast to a standard lecture was the ability to attract the lecturer's attention almost immediately by being able to visibly interrupt his or her presentation, letting a window pop up on his or her screen. This observation led directly to the implementation of the new feature for the lecturer to block clients, just to make sure that the lecture can be continued undisturbed.

Using the new interface instead of raising one's hand to request for asking was no problem at all for the students after a short introduction. In case of any doubts, like how to initiate a further inquiry, only a short and simple explanation was needed. Nevertheless, this type of user interface is not very intuitive; it should be improved in future work.

Another observation during our early tests concerned the acceptance of the questioner being recorded. Only few students asked how we deal with their personal rights and their data privacy, they were satisfied when they heard that these recordings would be accessible only by the lecturer and his or her research assistants and by the fellow students of their course, ensured by password protection and/or by being integrated into the Learning Management System (LMS) of our university.

Workload requirements for the lecturer

A new experience for the lecturer is the possibility of being interrupted during the lecture by the client window popping up. Similar to the experience we have observed with the students, it took some time getting used to it in order to not forget clicking on the "Now answering"-button when the lecturer starts to answer. Overall, the additional load for the lecturer is very small.

Another indicator for the small additional load is that lecturers do want their interface amended with additional features for the future, like the "block clients" check box, a switch board to manually steer the audience camera at certain seats, or a "Simulation/Animation" check box making sure that the slides are permanently on air. Even more, it is useful to evaluate whether such marks should be removed automatically

after a certain time. Finally, the user interface with clickable buttons is not very intuitive, as already mentioned for the student's user interface.

4.3.2. Experiences with the Event Reporting

The main task of the sensor tools module is to report events from sensor input to the virtual director. While the wired LAN connection from the Question Management Server to the virtual director module is very stable, the wireless LAN connections between the PDAs and the Question Management Server have more possibilities for disturbances.

The precision of the WLAN position estimation has been tested to be best using four to five access points (AP), and in our lecture hall two to three access-points are receivable depending on the client's location in the hall ensuring the WLAN connection to the university's network. Furthermore, four other APs were in reach used by different other institutes and researchers and have been switched off from time to time. All these APs cannot be controlled or removed by us.

In order to improve the reliability of the indoor positioning system, we wanted to increase the number of permanent receivable WLAN access points (AP) in the lecture hall, but we encountered problems if too many WLANs are active in a small area due to the channel selections of the already existing APs. If we used more than two additional APs for the WLAN positioning system and the system communication even the registering of the PDAs at the Question Management server failed.

We figured out this coherence by testing the setup in the lecture hall and for comparison in another part of our building with fewer APs in reach. Additionally, we ensured that one AP is sufficient for the communication with the PDAs. The system's logic controls that only one PDA is able to transmit and receive a significant amount of data at one time. Only during the registration process multiple PDAs may access the AP at the same time. But, as each registration process uses only six TCP packets there is not too much traffic. Additionally, we proved these theoretical thoughts in practice by having done repeatedly the successful registration of all used 40 PDAs at the Question Manager in less than one minute, outside the lecture hall where only our APs used the WLAN channels.

Thus we decided to set up only those two additional access points necessary for our distributed system, as shown in Figure 9, which are one AP for the communication of the lecturer's computer to the Internet and one AP for the communication of the PDAs with our system. When running only the necessary APs, the registering of the PDAs at the Question Management server was successful and stable, even in the lecture hall.

Another problem we encountered concerns the power consumption of the PDAs. In order to save energy, the PDAs put their WLAN cards into sleep mode after a specified time of no WLAN activity, and it takes from 100 ms up to approximately one second for the operating system to wake them up again when needed. This led at least to a delay when a questioner announced his or her question or even led to a complete drop out. To get the state of the PDA re-synced with the state registered at the Question Management server, either a simple click on the announce button or at least a reset of the PDA and a restart of the QM client was successful.

In case the announcement is successful, the newly opened UDP connection for audio streaming is enabled, and at least every minute a short state message is exchanged over TCP between the client and the server to keep the WLAN card awake. This continues as long as the Question Management software is either in the question or in the answering mode.

Also, we have exchanged the rechargeable batteries of the PDAs as they were already used for more than one year. In most cases, the new batteries were sufficient for the duration of a lecture (90 minutes) but depending on how many questions a student asked the life time of one battery charge may be even shorter. Fortunately, it is easy to simply replace a PDA during a lecture and register it with the running system to keep all students ready to ask questions.

Although the system works well in most cases, we thought of porting the PDA software to standard notebooks in order to overcome all problems which concern either the battery charge status or WLAN cards reacting too slowly. Porting of our software is future work.

4.3.3. Experiences with the Audio Streaming

Another task of the sensor tools module is one part of the virtual audio engineer. The QM client on the PDA samples the audio of the questioner and transmits it over UDP

using WLAN to the QM server from where it is fed into the video server of the audience to get encoded and streamed.

Tests with the PDAs have shown that sampling produced a significant load, and we had to figure out the trade-off between a small buffer size for short latencies and a larger buffer size not to overcharge the PDA. Finally, a ring buffer using four elements of 10,584 bytes each produced a jitter-free sound reproduction at a sample rate of 11.025 kHz with 16 bits per sample works perfectly.

The audio data we need to transmit over UDP is about 22,050 bytes per second which is equal to $22,050 \text{ bytes/s} \cdot 8 = 176,400 \text{ bit/s} = 176.4 \text{ kbit/s}$. The instructions sent over TCP only produce a small amount of data which does not carry much weight. Therefore, we at first tested to use an 802.11b WLAN with a gross bandwidth of 11Mbit/s, but unfortunately we encountered problems as the available net bandwidth for each user in reality is only a fraction of the gross bandwidth (about 50 percent under best circumstances). In addition, indoor reflections and overlapping channels of multiple APs in reach, as well as other devices using the 2.4 GHz band (like bluetooth devices) further decrease the available net bandwidth per device. Therefore, we decided to use at least a gross bandwidth of 54 Mbit/s as it is provided by the 802.11g protocol, for example. Fortunately all the participating devices supported 802.11g. Taking this bandwidth into account, we had no problems at all sending the sampled buffers over WLAN using UDP.

As UDP is a connectionless protocol, providing no error detection and no packet retransmission in case of packet loss, it is possible that single packets are dropped and do not reach the receiver; this actually occurs from time to time. Such a packet loss manifests itself in a break of 104 ms in the audio, as calculated in definition/formula 7 above.

Concerning the audio transmission, such small breaks are no big problem as they are short enough for a human user to interpret the missing phonemes, but we observed that the questioner gets irritated if his or her voice was additionally amplified for the lecture hall and slightly delayed due to the transmission. Nevertheless, the rest of the audience appreciated it as they could now easily understand the question.

As the delay mainly depends on the buffer size in the PDA and is directly linked to its performance, it may be a good idea for future work to either use faster PDAs or even port the client software for standard notebooks, which are much more powerful.

4.3.4. Overall Performance

The overall performance of the sensor tools module is good. It provides the necessary sensor input for the virtual director very reliably. In addition, the audio transmission works fine.

Nevertheless, there are some possibilities to improve the GUIs or to shorten the delay of the audio transmission, to name two aspects. One simple way of improving is to port the QM software client onto notebooks, which are used by many students anyway. Then it would be necessary to support different operating systems, as besides Microsoft Windows, Apple's MAC OS or Linux are often used by students. All of these topics can be addressed in future work, and will be described in more detail in the Summary chapter.

4.4. Experiences with the AV Mixer/Recorder

The AV Mixer/Recorder module is one of the crucial parts of the prototype of our distributed Automatic Lecture Recording system. In contrast to the virtual director module which is the core component from the scientific point of view, the AV Mixer/Recorder is the core component from the craftsmanship point of view.

As the AV Mixer/Recorder deals with many uncompressed audio and video streams in parallel, it is hard but very important to obtain the real time capabilities. In particular, we had to overcome some constraints of the programming language we used which will be presented in the following sections.

4.4.1. Experience with the AV Decoding

All audio and video data reach the AV Mixer/Recorder using RTP streams started by a preceding RTSP negotiation between AV servers and clients. For each audio and each video stream, a separate RTP stream is used while the according pairs of audio and video streams get synchronized by RTSP messages.

We use the FFmpeg Dynamic Link Libraries (DLL) (FFmpeg, 2009) called by the TAO.FFmpeg API (TAO-Framework, 2009) for negotiating, receiving, and decoding

the AV streams. The results are fully decoded audio and video data streams. While the audio data is returned as raw PCM sample sets with 8 kHz at 16 bit quantization, the video data is returned as an *AVFrame* object containing the YUV data of each frame with a 4:2:2 color sub-sampling. Before any further processing takes place, we convert this *AVFrame* object into an RGB bitmap object.

Curiously, we found out that the video data is not necessarily served at the frame rate of 25 fps we defined in the AV server hardware set-up. All frames transmitted do have a correct timestamp to keep the audio and the video streams synchronized, but depending on the load the AV server has, the frame rate goes down to e.g., 22, 20, or even 7 frames per second. However, the audio data is constantly sent with 25 fps at the correct data rate. After we checked our network set-up, and this strange behavior continued to occur even in a directly connected peer-to-peer network, we became sure that packet losses can not be the reason for it. So, we inquired at the manufacturer of the AV servers about this phenomenon and he confirmed the behavior as the AV servers are able to provide “up to 25 fps” and that “this value can be reached only under perfect circumstances; otherwise the frame rate will be decreased automatically”.

We implemented a work-around as we needed to provide precisely 25 fps for the final video: as soon as an audio frame or a video frame is completely decoded it is filled into a buffer. Completely independent from the filling of the buffers, only the buffers needed for the active shot get read out by another thread every 40 ms to reach the frame rate of 25 fps. Of course, the parallel access to the buffers from different threads is synchronized using the monitor concept. To fill the buffers, we use events every time the audio or video data is fully decoded, while the readout is controlled by a recurring timer. As all threads accessing the buffers are completely independent from each other, the OS is able to distribute them on different central processing unit (CPU) cores.

4.4.2. Experiences with the Output Trigger

At first, we have built the recurring timer we need to provide a precise frame rate of 25 fps by using the standard timer of C#. Unfortunately, our tests revealed that this timer triggers not after 40 ms but after about 50 to 60 ms. This behavior has been confirmed by the Microsoft Developer Network (MSDN) library in which the actual resolution of the two standard objects *System.Timers.Timer* and *Sys-*

tem.Windows.Forms.Timer is 55 ms. Strangely, this is true even if the parameter can be used to set the interval to 1 ms.

We found the possibility to use the so-called Win32 Multimedia Timer Functions out of the WINMM-DLL which is able to provide a precision of about 1 to 3 ms. So, we were able to read out the buffers precisely every 40 ms.

4.4.3. Experiences with the Video Processing

Video processing here is image processing frame by frame. The basic functionality of the AV Mixer/Recorder consists of three features: transition from shot to shot using a hard cut, transitions by cross-fading, and the picture-in-picture effect used.

Each special effect, dissolving or picture-in-picture of only two standard video sources, produces some load on modern computers but still is manageable. However, the load increases significantly if four standard video sources have to be available all the time out of which two can be selected for an effect. The maximum load would be generated if the four sources are at first combined to two picture-in-picture sources and dissolved afterwards.

The output of a frame is triggered by the multimedia timer as mentioned above. Therefore, calculating the final frame has to be done in one single thread. If only one basic video stream is selected, the readout bitmap is just copied to the output. If a PiP-image is requested, it has to be created in real-time out of the two basic video streams. If a dissolve is requested at first the current percentage of the two bitmaps has to be calculated for the dissolve. For example, a dissolve which lasts two seconds uses 50 frames, i.e., with each frame the transparency of the final image decreases by two percent. Then, the starting image and the partly transparent final image have to get combined to one image.

It is obvious that cross-fading two PiP images produces the highest load in our scenario. As it is done in one single task, it is not possible to distribute it over multiple CPU cores. In general, it is questionable whether it would be feasible trying to distribute it over multiple CPU cores as all used bitmaps must be accessible to all participating cores. Even more, it is a question of the trade-off between the efforts of distributing data for parallel computing versus the effort of computing the result on a single core.

As any calculation for the final output frame has to be done in real-time, we need a very powerful multi-core computer like the Dual-Xeon Quad-Core at 2.66 GHz used for our prototype.

4.4.4. Experience with the AV Output

Up to now, we were able to keep up with the fundamental prerequisite of doing everything in real-time. The final step in the process chain of the AV Mixer/Recorder is saving the resulting audio and video streams into a video file. Alternatively, we could hand them over to a streaming server.

As already mentioned, we tried many different ways of encoding the frames into a video stream and save them to a file in real-time. But encoding and saving was not successful, neither by writing AVI files employing the AVIFIL32.DLL (due to the limited file size of AVI 1.0 files of 2 GB), nor by using the TAO framework on top of FFmpeg due to its complexity to set the correct parameters and memory access for AVI files, nor by using the Quicktime encoder due to the wrong *Single-Threaded-Apartment model* which additionally to the already mentioned disadvantages does not allow to use the Windows clipboard functionality as needed. Finally, there were two possibilities left to get the frames to the hard disk as a video file: either to write an input filter for the Microsoft DirectShow filter graph or to save each single frame to hard disk and join them to a video file later on.

Implementing the first alternative was impossible for time constraint reasons. Thus we implemented the second alternative as a work-around and deferred the full real-time capability of the entire virtual camera team to future work. Nevertheless, even the simple task of writing frames to hard disk has its perfidy. Saving the bitmap objects to disk in Bitmap file format (BMP) was not fast enough to keep up with the frame rate of 25 fps even as we employed a Redundant Array of Independent Disks (RAID) level 0 system. The reason is the large file size of BMP files. We found that saving JPEG files encoding the bitmap objects was fast enough.

The next drawback we encountered was the behavior of the Microsoft Windows file system when administrating a large number of files in a single directory. The file system gets slower and slower with every single file added to this directory. As we have to save about 90 minutes of lecture video, corresponding to 5,400 seconds which is 135,000 frames at a frame rate of 25 fps, we experienced saving drop outs as soon we

exceeded a number of about 30,000 frames. As this is a built-in behavior of Microsoft Windows, we had to implement a work-around: we distributed the saved files over multiple directories so that each directory does not contain more than 22,500 files, corresponding to 15 minutes of video. Now, the file system works sufficiently fast.

4.4.5. Overall Performance

The overall performance of the AV Mixer/Recorder is fairly good. It processes all necessary tasks in real-time and in a robust and stable manner but of course it needs a powerful computer.

Just the way of saving the final video to hard disk should be improved, most suitably using a DirectShow input filter which promises the best flexibility and capability for Windows XP.

Having a short outlook on future work, two main things could significantly improve the AV Mixer/Recorder: at first, optimizing the parallel processing of threads and optimizing the encoding and saving of videos to disk. Unfortunately, the distribution of threads on multiple CPU cores was completely dependent of the capabilities of the OS up to the time we did our research using Microsoft's Visual Studio 2005. The distribution of threads could not be controlled manually, and its add-on for providing such functionality was not yet stable at all. At second, Microsoft introduced a new framework called *Windows Media Foundation* for processing videos, amongst others inside Windows Vista. It significantly simplifies the complexity of developing media processing components compared to the older DirectShow framework.

4.5. Experience with the Sound Engineer module

The virtual sound engineer module is separated into two parts, one is included in the question management client on the PDAs and the second in the AV Mixer/Recorder. As we already described our experiences with the part on the PDAs, we are now focusing on the part inside the AV Mixer/Recorder.

4.5.1. Audio Mixing and Mastering

In our prototype setup, we get up to three audio streams from the AV servers of the lecturer, of the slide PC, and of the audience. As we have routed the audio of the lecturer into his or her presentation computer and combined it with any sounds of anima-

tions or simulations of the computer, we only use the slides audio stream for the pre-mixed audio of the lecturer and the computer.

All audio coming from the questioner via the QM client and the QM server is transmitted via the audience audio stream. So, we only have to mix two different audio streams in the AV Mixer/Recorder. Nevertheless, all algorithms are built for processing all three audio sources as this makes no big difference in the resulting load.

As already mentioned in Section 3.4, the algorithms of applying the noise gate for normalization, for re-sampling and for mixing are robust, and work in real-time. While the function of all algorithms except normalizing the volume is based on an on-sample base, i.e., the smallest unit to operate on, are only one or two samples. In contrast, normalizing operates only on sample sets of audio. The reason is that the algorithm has to find the loudest sample in the sample set of audio to determine the factor by which the whole sample set gets amplified. If the selected part is too small the algorithm only takes a local maximum into account which is not representative for the entire recording. Therefore, normalizing is mostly done as the last step during mastering, taking the whole recording into account in order to use the global maximum.

As we have to bring all audio streams to the same volume level before mixing them, we have to amplify them if they do not contain only silence. There are two possibilities to amplify signals in order to achieve comparable volume levels:

1. Normalizing, i.e., determine the loudest sample, calculate the factor to bring this loudest sample to a defined volume level, and amplify the whole sample set with this factor,
2. Compressing/limiting, i.e., define a characteristic curve with different amplification factors, dependent on the input volume level. Figure 34 shows such a curve as an example.

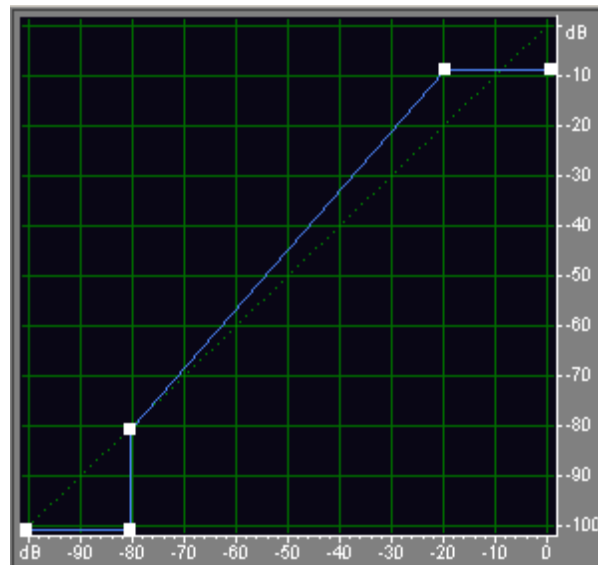


Figure 34: Screen shot of an exemplary curve of a combined noise-gate, compressor, and limiter.
Input-dB on the x-axis, output-dB on the y-axis.

The horizontal axis shows the input signal strength in [dB] while the vertical axis shows the resulting output signal strength in [dB]. The line following the first bisection is the neutral element for a compressor/limiter; it does not change the amplification. To positively amplify a signal, the line must be above the first bisection while a line below diminishes a signal. The example in Figure 34 shows a “noise gate” for the input signal strengths of -100 dB to -80 dB, so these input levels are mapped to -100 dB for the output signal strength. In the range of -80 dB to -20 dB for the input signal strengths, a linearly increasing amplification takes place, mapping them to the range of -80 dB to -9 dB. This part “compresses” the input signal. Input signal strengths in the range of -20 dB to 0 dB gets “hard limited”, i.e., strictly mapped to the output signal strength of -9 dB. The changes between noise gate and compression and between compression and limitation are done in this example by so called “hard knees” which change the behavior in a very abrupt way. In contrast, there is the so-called “soft knee” rounding the corners, and therefore the transfer from one mode to the other is less aggressive.

The advantages of a compressor/limiter algorithm are that it can operate on an on-sample base and that it can combine many different tasks easily into one processing step, e.g., “noise-gating”, “compressing”, and “limiting”. The disadvantage is that the algorithm is much more complex than normalizing, and we cannot implement and test

it successfully due to the time constraints we encountered. Nevertheless, it is planned for future work.

For our prototype, we amplify the audio data streams using the normalizing algorithm. As it needs to operate on a sample set, we need to define a useful one. At first glance, it may be useful to process the sample set of one audio frame of 40 ms at once. Unfortunately, if we normalize these small parts of audio data, every part will be amplified with a different factor, leading to block artefacts at the transition from the end of one part to the beginning of the next part. Due to the different amplifications, the slope of the curve changes significantly in a very short time, leading to a clicking noise. As such noises occur repeatedly every 40 ms, the whole recording is spoiled by cracklings, making the entire processing unusable. Figure 35 shows two 440 Hz sine curves, one with different amplifications and one without. Inside the red mark, there is the source of the clicking noise.

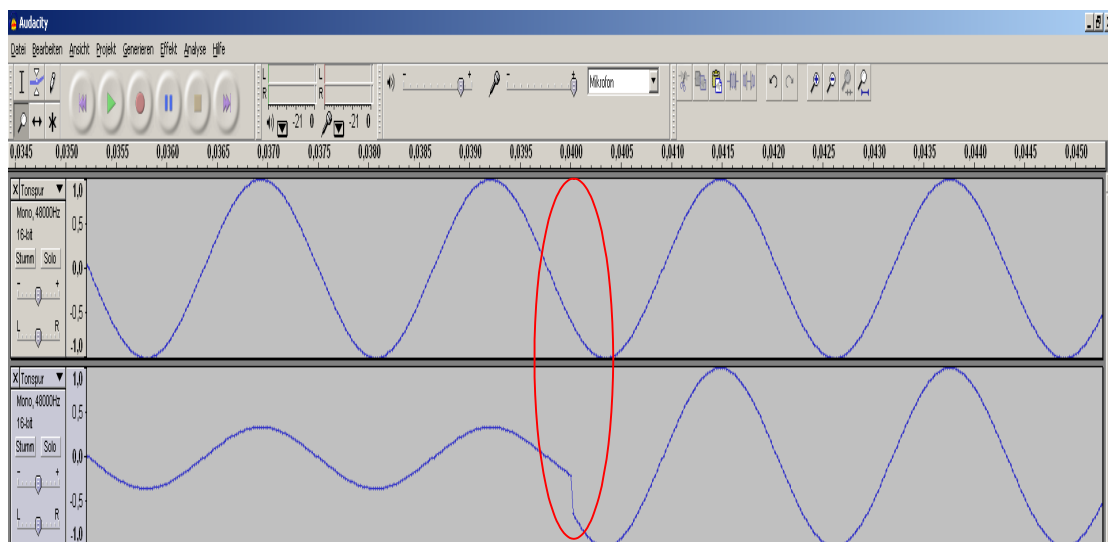


Figure 35: A 440 Hz sine curve with different amplification factors in adjacent audio frames.

The next way of defining a useful sample set is to use the entire recording at once. So, a global maximum can be taken into account which normally leads to very good results. This is true for sources with continuous audio signals and without any silence in it. Due to the question – answer interaction of questioner and lecturer, we anticipate silent parts in the audio data streams which can easily lead to long parts of amplified silence producing significant noise. Thus, this way is not optimal for our prototype. This is a consequence of the way we implemented the noise gate: it checks for silence in the whole sample set and if only silence is found the entire sample set is neglected.

But it does not check for silent parts inside the sample set in order to keep it simple and fast. Besides this theoretical drawback, we would have to face memory allocation problems when loading three audio streams of 90 minutes into the memory in order to find the global maximum level in the audio streams.

Thus, we need to find a compromise to find amplification factors based on a significant amount of the data, as well as to suppress noise as well as possible. We found this compromise in the following approach: we save all incoming audio data directly onto the hard disk and process it afterwards producing a single WAV-file. This file is used as the final sound track for the video file. This reduces the load of the AV Mixer/Recorder significantly but still enables us to select any useful sample set size for normalizing. During our tests, we observed the duration of one second to be a useful sample set size as it leads to similar amplification factors and therefore to rare clicking noises. Another consequence is that all durations of amplified silence occurring before a questioner asks are shorter than one second which is noticeable but not very disturbing.

4.5.2. Overall Performance

The overall performance of the virtual sound engineer is good, and all algorithms are fast enough to process the audio data in real-time. Due to the already mentioned time constraints and CPU requirements of the AV Mixer/Recorder, we had to implement a work-around to prove our concept, which we successfully did. So, it fits perfectly into our virtual camera team.

In addition, we pointed out the planned future work to optimize the virtual sound engineer; it will be a perfect supplement to future work on the AV Mixer/Recorder.

4.5.3. Improving the AV Mixer/Recorder

In future work the AV Mixer/Recorder should be capable of live streaming. There are two parts necessary to improve to achieve this goal. At first the audio normalizing algorithm has to be exchanged by a real time capable algorithm like the dynamic volume adjustment algorithm mentioned in Chapter 4.5.1. The source code of this noise-gate, compressor, and limiter is included in the prototype, but has not yet been tested at all. As shown in Figure 34, a function is defined by an arbitrary number of nodes,

mapping an input volume given in [dB] to an output volume in [dB]. The way to implement this approach in an algorithm consists of four steps:

1. providing converting algorithms of sample values to [dB] and vice versa,
2. computing the amplitudes of the sample set in real-time,
3. computing the amplification factors for every single sample value in the sample set based on the given characteristic curve of the defined function, and
4. multiplying the original sample values with the computed amplification factors.

The following function shows the main routine while the functions called by it can be found in Chapters 7.2.3 to 7.2.7 in the Appendix.

```
public Int16[] NoiseGateExpanderCompressorLimiter
    (Int16[] myPCM, double[, ] CompressionLine)
{
    Int16[] result = new Int16[myPCM.Length];
    Int16[] extrema = new Int16[myPCM.Length];
    double factor = 0;

    // calculate values "riding" on the local maxima of the samples
    extrema = WaveExtrema(myPCM, carryExtremaOver);

    // used to implement a smooth transition from one sampleset to the next
    carryExtremaOver = extrema[extrema.Length - 1];

    // transform DezibelIn into ampFactors and
    // multiply Samples with ampFactors
    for (int count = 0; count < myPCM.Length; count++)
    {
        factor = getFactor(extrema[count], CompressionLine);
        result[count] = (Int16)Math.Round(myPCM[count] * factor);
    }
    return result;
}
```

The way of converting sample values to [dB] and back is based on the application note 1MA98 from Rohde & Schwarz (Rohde & Schwarz, 2006) and is realized in the functions “*Sample2DB*” and “*DB2SampleValue*”. The function used to determine the input values has to “ride” on top of the amplitudes of the sample set. While different implementations are possible, we decided to use the following approach: At first, only the absolute values of the sample set are taken which means that any negative values get multiplied with -1. In the second step, each local maximum of the wave curve is stored as the next amplitude value to “ride on”, see function “*WaveExtrema*”. These

amplitude values are converted into [db] for input, mapped to the corresponding output [dB] defined by the function “*DefineCompressionLine*” and using a linear interpolation, and converted back into sample values. An expander–compressor–limiter can either use linear interpolation for the values between the given nodes or use spline interpolation. Now, it is easy to determine the amplification factor to bring the original sample value to the desired amplitude (function “*getFactor*”). In order to achieve smooth transitions from one sample set to the next, we use the variable “*carryExtremaOver*”. Before the first run of the function “*WaveExtrema*”, it is set to zero, and after each further run it is set to the last value of the “*extrema*” array. Thus, it is easy to successfully apply this routine to any incoming audio sample set which consists of 240 samples per frame.

The second part, necessary to enable live streaming, is a “DirectShow Source Filter” accepting bitmaps as an input as mentioned in Chapter 4.4.4. *DirectShow* is the preferred way to process audio and video streams in the Microsoft Windows XP operating systems, like the Windows XP professional 32 bit operating system we used. There are three kinds of filters: source filters, transform filters and AV renderers.

For tasks like multiplexing AV streams and displaying the video streams or making the audio streams audible, filters are available. In contrast, filters to send the streams to the network and to receive them from it, e.g., by using the RTP protocol, can either be purchased or have to be self-developed, e.g., based on the open-source live555-libraries available from (live555, 2009). As these libraries were originally developed for Linux operating systems, there is some porting effort necessary to successfully compile the libraries using Microsoft Visual Studio 2005. In addition, the libraries have to get adapted to the rigorous specifications that the *DirectShow* filter framework requires.

A similar situation exists for *DirectShow* source filters accepting bitmap objects and/or raw PCM data arrays as their input. These filters have to be developed from scratch only with basic support from the *DirectShow* Filter Development tutorial (DirectShowTutorial, 2009) which is based on the Windows SDK (WindowsSDK, 2009).

As mentioned, developing such filters was not possible with the time constraints we had, and therefore enabling live streaming is still subject to future work.

5. Evaluation with Students

After having the prototype of the distributed Automatic Lecture Recording system ready for use, we set up an evaluation study analyzing the impact of two different ways of lecture recordings. The first type of lecture recording is the traditional version, recording the slides and the lecturer's audio. The second type is based on the prototype of the distributed Automatic Lecture Recording system, recording videos of the lecturer, the slides, the audience and an overview shot, as well as the audio of the lecturer, his or her computer, and questioners out of the audience.

In the following sections, we describe the evaluation study in detail, its design, realization, and its results.

5.1. Evaluation Description

We want to evaluate the two types of lecture recordings concerning the *fascination/interest of the students, motivation, and learning gain*.

As we want to evaluate the lecture recordings but not the lecture itself, we decided to record a basic lecture of a topic not in the curricula of our university but held by me for another university. The main advantage is that no previous knowledge was necessary for the lecture, and therefore students from every course of studies can participate. The topic of the recorded lecture was "Audio recording and audio cut for video productions".

5.1.1. Evaluation Design

The evaluation was planned to be completed in one week. The design of the evaluation study was developed in close cooperation with our colleagues from the Chair of Education of the University of Mannheim, in particular with Dr. Tanja Mangold and Professor Dr. Peter Drewek whose assistance we gratefully acknowledge.

On the first day we held a lecture which was recorded simultaneously by both types of lecture recordings: we got two versions containing exactly the same topic but with significant differences in the recording itself. While the first version constantly showed the slides, the second version switched between different shots of the lecturer,

the slides, the audience, and the overview shot, as well as medium close shots of questioners out of the audience, as directed by the prototype of our virtual director module.

In the next step, the participants of the study chose one of three days to participate by watching the different lecture recordings. On the first day, we showed the first version which was the standard lecture recording. On the second day, we showed the new version recorded by the prototype of the Automatic Lecture Recording system. On the third day, we showed both videos simultaneously but in different rooms. We randomly divided the participants' group of the third day into two subgroups and showed the first version of the video to the first subgroup and the second version to the second. Thus, we were able to balance the number of participants for each version of the video to be nearly equal.

The sample itself was done in four steps. The first step was a pre-test in order to get to know the already available knowledge of the participants. As already mentioned, no knowledge was necessary for this lecture but of course it may exist, therefore we tested it. Naturally, the topic of the test refers to the content of the lecture. Figure 36 shows a translated version of the test while the original test in German is shown in the Appendix in Section 7.3. The test consists of fourteen questions, a total of 20 points can be reached.

As we also want to determine the learning gain achieved by watching the lecture recordings, we also did a post-test afterwards, asking the same questions as in the pre-test. The difference of the achieved points represents the individual learning gain of the attendees.

<div style="display: flex; justify-content: space-between; font-size: 0.8em; margin-bottom: 10px;"> <div>FAKULTÄT FÜR SOZIALWISSENSCHAFTEN Lehrstuhl für Erziehungswissenschaften I Prof. Dr. Peter Drewek</div> <div>INSTITUT FÜR INFORMATIK Lehrstuhl für Praktische Informatik IV Prof. Dr. Wolfgang Effelsberg</div> <div>UNIVERSITÄT MANNHEIM</div> </div> <table border="1" style="margin: 0 auto; border-collapse: collapse; text-align: center;"> <tr> <td style="padding: 2px 5px;">Video-#a</td> <td style="padding: 2px 5px;">VPW-#a</td> </tr> </table> <p>Pre-Test to the video</p> <p>The answers of the following questions are only used to evaluate your knowledge in the area of "audio recording and audio cut for video productions" and therefore this survey is done <i>anonymously</i>.</p> <ol style="list-style-type: none"> 1. (2p) Describe the difference between sampling-depth and sample-rate? 2. (1p) Name the unit in which the sampling-depth gets measured? 3. (3p) Name the typical parameters of high-quality digital video. 4. (2p) Name two different characteristics of microphones. 5. (1p) What do capacitor-microphones need to work? 6. (2p) Describe the difference between a pop-filter and a wind-screen? 7. (1p) Name the number of persons which should be interviewed at once with one microphone at most? 	Video-#a	VPW-#a	<ol style="list-style-type: none"> 8. (2p) Describe the difference between a microphone boom pole and a microphone stand? 9. (1p) What is an „atmosphere“-track? 10. (1p) Describe the function of an „atmosphere“-track? 11. (1p) Name the mentioned effect, which may appear if a signal gets recorded by multiple microphones? 12. (1p) Which should be the loudest dB mark of a recorded sound? (Unit: dB) 13. (1p) Which dB mark should be reached at least for a quiet but well-audible recorded sound? (Unit: dB) 14. (1p) Describe an example in which a dissolve of two consecutive audio clips is more reasonable than a hard cut?
Video-#a	VPW-#a		

Figure 36: Translated knowledge pre-test.

The second step in our evaluation design consisted of presenting the lecture recording. As the maximum duration of one lecture is limited to 90 minutes in Germany, the actual durations of the lecture recordings was 81 minutes and four seconds for the first, standard version, and 81 minutes and 11 seconds for the second, enhanced version. The difference of seven seconds comes from manually starting and stopping both recordings but does not lead to any difference in the content as only the first and the last shots differ minimally in their duration.

The recorded videos differed in the following details:

Table 11: Differences between the tested two types of lecture recording.

	Standard Lecture Recording Video	Enhanced Lecture Recording Video
Audio of questioners re- corded?	NO	YES
Audio of lecturer re- corded?	YES	YES
Audio of simula- tions/animations recorded?	YES	YES
Talking head of lecturer recorded?	NO	YES
Video of the audience or of questioners recorded?	NO	YES
Overview shot recorded?	NO	YES
Slides recorded?	YES	YES
Switching between differ- ent views?	NO	YES
PiP of two sources avail- able?	NO	YES

In the third step of our evaluation design, the attendees had to fill out a questionnaire asking about the participants' motivation and interest. This questionnaire consisted of three different sections, each investigating one parameter, so-called "construct", namely *attentiveness*, *comparability of the video types*, and *motivation*. While the first two constructs focus on one aspect of fascination/interest each, the last one focuses on four factors, namely *anxiety (of failure)*, *probability of success*, *interest*, and *challenge*, in order to give an answer about the motivation. Each factor is tested by several

items: some of them were recoded during the analysis to fit numerically into the evaluation process. Figures 37 to 38 show the translated questionnaire.

FAKULTÄT FÜR SOZIALWISSENSCHAFTEN
Lehrstuhl für Erziehungswissenschaft I
Prof. Dr. Peter Drewek

INSTITUT FÜR INFORMATIK
Lehrstuhl für Praktische Informatik IV
Prof. Dr. Wolfgang Ertlberg

UNIVERSITÄT
MANNHEIM

Video-#a VPH-#a

Questionnaire to the lecture recording video

We are interested in **your opinion concerning the lecture recording video** you have just watched. We would like to ask you some questions. These questions are **no test**, thus there are no correct and no wrong answers. This survey is done **anonymously**, therefore we would like to ask you to answer these questions in all conscience.

Please take care during filling in the questionnaire that:

- There are multiple answers possible for most of the questions, out of which **always only one answer** that fits your opinion should get ticked. Please **tick only one answer per line**.
- It is important to us to get to know personal opinions; Therefore, please **fill in** the questionnaire **solely by yourself**.

Demographic data:

(01) How old are you?
I am _____ years old.

(02) Please tick your gender?
male ☐ female ☐

(03) What is the name of your course of studies?

(04) What is the number of your current semester?
The number of my current semester is _____.

We would like to know your **current attitude** concerning the video watched. Therefore, you find some statements on this page. Please tick the field that describes your attitude best. Please make sure to **tick only one field per line**.

	Does not apply at all						Does fully apply
(5) a) I like this kind of video. (I)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
b) I believe that using this video I am prepared to successfully answer questions concerning this lecture. (P)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
c) Presumably, I am not able to successfully answer content-based questions concerning this lecture. (P)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
d) I felt myself under pressure paying attention to this video. (A)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
e) This video was a real challenge for me. (C)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
f) After having watched the video, the content seems to be very interesting. (I)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
g) I am looking forward how well I have paid attention to this video. (C)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
h) I am a little afraid to disgrace myself when trying to answer questions concerning the content of the video. (A)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
i) I was fully intended on working hard at this video. (C)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
j) I need no reward watching a video like this. It is fun. (I)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
k) It embarrasses me to fail answering content-based questions of this video. (A)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
l) In my opinion, everyone can learn with the help of this video. (P)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
m) I believe I cannot learn using this video. (P)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
n) If I learn anything using this video, I am proud of my capability. (C)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
o) I am a little concerned thinking of this video. (A)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
p) Even in my spare time I would occupy myself with this kind of video. (I)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
q) The performance demands coming from this video paralyze me. (A)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Figure 37: Pages 1 and 2 of the translated questionnaire.

Please remember the video watched. The next questions do **not refer to the content of the video** but we would like to know how you felt watching **this type of video**. Please tick the field that fits your opinion best for each of the following statements.

	Does not apply at all					Does fully apply
(6) a) I was able to attentively follow the video's content.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
b) I have been easily distracted from watching the video.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
c) I occupied myself with other tasks during the runtime of video.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
d) The video was tedious to me.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
e) I kept focused while watching the video.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
f) During the runtime of the video I did not feel any boredom.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Please imagine that the courses you attend and you are going for examination are recorded in the way of the video and provided online for download. Please tick one field per statement that fits your opinion best.

	Does not apply at all					Does fully apply
(7) a) I cannot imagine learning for examination using this kind of video.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
b) I would rather attend the lecture than learning to use this kind of video.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
c) I would like to use this kind of video as an additional medium to the lecture.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
d) A lecture cannot be replaced by this kind of video.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
e) I would appreciate using this kind of video instead the lectures taking place.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
f) I only would use this kind of video, if I were prevented for any reason from attending the lecture.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

In the following we are interested in your reason watching this lecture recording video. Please tick one field per statement that fits your opinion best.

	Does not apply at all					Does fully apply
(8) I watched this video because ...	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
a) ... I have been paid for attending.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
b) ... I wanted to support the research in this area by attending.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
c) ... its content is important for my future career.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
d) ... I am interested in the technology presenting the learning matter.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
e) ... I like learning multimedia-based.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Finally, we are interested in your general opinion and statement concerning this video.

	1 time	2 to 3 times	4 to 5 times	More than 5 times
(9) a) How many times do you think you have to watch this video in order to perceive the entire learning matter of the lecture?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
b) According to your opinion, how much did you learn from this lecture by this kind of video direction?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

(10) Do you like to add a comment or to give us a message?

□

Figure 38: Pages 3 and 4 of the translated questionnaire.

The fourth step in our study was the post-test. It evaluates the knowledge the participants have after watching one of the lecture recording videos. As mentioned, the German original can be found in Appendix 7.3.

5.1.2. Description of the sample and its participants

For our evaluation study, we recruited students of our university. We had 33 participants in total, out of which 30 (90.9 %) came from the study course “teachers at secondary schools” (Lehramt am Gymnasium). The rest were students, too, but from different study courses, namely one from “psychology” (3.03 %), one from “business administration” (3.03 %) and one from “social science” (3.03 %). We divided the participants nearly equally into both groups. The first version of the video was viewed by 17 attendees (51.5 %), while the second version was viewed by 16 people (48.5 %). They consisted of 25 women (75.8 %) and 8 men (24.2 %).

They were between their second and their thirteenth semester, their age varied from 20 to 37 years. The average age was 24.3 years with a standard deviation of 4.202 and a median of 22.0 years.

5.1.3. Operationalization of the constructs

Each of the following constructs consists of one or more groups of items belonging to one aspect of the investigation. In order to weigh the aspects and their items correctly, we needed a number of parameters providing the necessary information. Besides statistic parameters like the average, the standard deviation and the underlying distribution, we used special parameters describing, e.g., the selectivity of the items and the reliability of the constructs. We used the definitions and statements of (Bortz & Döring, 2006), pages 196 to 221.

By **analyzing the factors of the test**, the *dimensionality*, the *selectivity*, and the *homogeneity of its items* can be determined.

The *dimensionality* provides the information whether the items of a test cover only one construct or some (sub-) constructs. If it covers only one construct, we will speak of a one-dimensional test. If it covers multiple constructs or sub-constructs, a multi-dimensional test is very likely. In order to prove the dimensionality of a test, a *confir-*

matoric factor analysis is needed which produces a so called *factor-charge* for each single item.

If all single-item inter-correlations of one test highly correlate with one single general factor they can therefore be reduced to it, achieving a one-dimensional test, which refines the theoretic assumptions. In contrast, the item-inter-correlations highly correlate with multiple factors when having a multi-dimensional test. Therefore, the quality of multi-dimensional tests should be investigated separately by their factors.

The *selectivity* (r_{it}) is defined as the correlation of the answer of this item to the answer of the construct. It shows to which extent a single item represents the overall result of the construct. The domain of the selectivity coefficient lies in the range from minus one to plus one. According to (Bortz & Döring, 2006), values from 0.3 to 0.5 should be interpreted as good while values from 0.5 to 0.7 should be interpreted as very good. Values below 0.3 occur with items having nothing in common while values above 0.7 arise when having items testing identical aspects. The wording of an item is chosen in such a way that no negative values of correlation can appear. This process is called re-coding of the items. It eases the comparison and further processing of the values. Items of a construct having a lesser selectivity may describe another dimension, may distract the focus and therefore get removed from the construct.

The *homogeneity* is a measure for the average correlation of the items of one construct to all other items of a test. Concerning the estimation of reliabilities, the average item inter-correlation goes into the alpha coefficient of Cronbach, sometimes also called *index of homogeneity*.

Analysis of the reliability

The *reliability* is a characteristic measure for the degree of precision of the investigated parameter. Its coefficient lies in the range from zero to one where a value of zero means that the measurement consists of errors only. In contrast, a coefficient value of one means that the measured value is identical to the real value. For analyzing our evaluation study, we use the alpha coefficient of Cronbach as our coefficient of reliability.

According to (Bortz & Döring, 2006) for non-explorative intentions, the reliability values for moderate constructs are in a range from 0.7 to 0.8, for very good constructs between 0.8 and 0.9.

In order to be able to correctly judge the values of the alpha coefficient, we have to keep in mind that the alpha value raises higher the more items are in the scale and the higher the item inter-correlation values are. If a construct is not one-dimensional and/or its alpha coefficient of Cronbach is bad the corresponding item gets removed in order to achieve a one-dimensional scale.

However, as our evaluation study has an explorative characteristic, constructs having a reliability value above 0.7 are considered to be very good constructs.

Fascination / Interest

The first term we wanted to evaluate was “fascination” or “interest” in natural language. From the scientific point of view, we defined two constructs describing the aspects more precisely: *attentiveness* and *interest in the video*. Both constructs were created from scratch by Dr. Tanja Mangold for this evaluation study. Her support is much appreciated.

The scale of the construct *attentiveness* was quad-staged and reached from “*does not apply at all*” to “*fully applies*”. The construct consists of five items two of which were recoded. All values are ordered so that the higher the average value is, the higher the attentiveness. The results of the analysis of the factors show that the construct is one-dimensional. Its alpha Cronbach value of 0.83 proves that the construct is reliable. The average value of the entire construct is 2.35, and it has a standard deviation of 0.689.

The coefficients of selectivity, the average values and the standard deviations of the items are shown in Table 12.

Table 12: Items of the construct "attentiveness" (T. Mangold).

Item	r_{it}	Average	Std. Dev.
I was able to attentively follow the video's content.	0.501	2.58	0.936
I have been easily distracted from watching the video. (recoded)	0.451	2.76	0.902
The video was tedious to me. (recoded)	0.509	1.91	0.879
I kept focused while watching the video.	0.397	2.76	0.902
During the runtime of the video, I did not feel any boredom.	0.423	1.76	0.830

The construct *interest in the video* consists of four items. They were all recoded in order to achieve ordered values so that the higher the values are the better our type of lecture recording video is rated. The results of the analysis of the factors show that the construct is one-dimensional. Its alpha-Cronbach value of 0.83 proves that the construct is reliable. The average value of the entire construct is 2.05, and it has a standard deviation of 0.881.

The coefficients of selectivity, the average values, and the standard deviations of the items are shown in Table 13.

Table 13: Items of the construct "interest in the video" (T. Mangold).

Item	r_{it}	Average	Std. Dev.
I cannot imagine learning for examination using this kind of video. (recoded)	0.426	2.55	1.15
I would rather attend the lecture than learning to use this kind of video. (recoded)	0.669	1.82	1.04
A lecture cannot be replaced by this kind of video. (recoded)	0.632	1.91	1.02
I only would use this kind of video, if I were prevented for any reason from attending the lecture. (recoded)	0.382	1.94	1.09

Motivation

The second term we wanted to evaluate was *motivation*. This term defines a precise aspect in normal language as well as a construct. In order to survey this construct, we fall back on the work of (Rheinberg, Vollmeyer & Burns, 2001). They developed a *Questionnaire to assess Current Motivation* in learning situations (QCM). It has to be adapted to the different fields of application. Tanja Mangold thus revised the items of the construct. It consists of four factors, namely *interest*, *probability of success*, *anxiety*, and *challenge*. Its scale is seven-staged from “*does not apply at all*” to “*fully applies*”. The higher the average value is, the higher is the interest, the probability of success, the anxiety, respectively the challenge.

The *anxiety* is a one-dimensional, one-factorial construct, having a Cronbach alpha value of 0.78, an average value of 3.20, and a standard deviation of 1.50. The *probability of success* is a one-dimensional, one-factorial construct, having a Cronbach alpha value of 0.86, an average value of 4.08, and a standard deviation of 1.552. The *interest* is a one-dimensional, one-factorial construct, having a Cronbach alpha value of 0.84, an average value of 3.11, and a standard deviation of 1.32. In contrast, all four items of the construct *challenge* charge evenly onto different factors. The construct

challenge is therefore not reliable and thus we waived the evaluation of the partial scale of it.

The coefficients of selectivity, the average values and the standard deviations of the items are shown in Table 14.

Table 14: Items of the construct "motivation" (T. Mangold based on Rheinberg, Vollmeyer & Burns, 2001).

Item	r_{it}	Average	Std. Dev.
I like this kind of video. (I)	0.549	3.58	1.678
After having watched the video, the content seems to be very interesting. (I)	0.472	3.52	1.584
I need no reward watching a video like this. It is fun. (I)	0.427	2.97	1.591
Even in my spare time I would occupy myself with this kind of video. (I)	0.424	2.36	1.537
I believe that using this video I am prepared to successfully answer questions concerning this lecture. (P)	0.676	4.12	1.47
Presumably, I am not able to successfully answer content-based questions concerning this lecture. (P)	0.711	4.30	1.85
In my opinion, everyone can learn with the help of this video. (P)	0.784	3.73	1.86
I believe I cannot learn using this video. (P)	0.781	4.15	2.14
I felt myself under pressure paying attention to this video. (A)	0.284	3.82	1.88
I am a little afraid to disgrace myself when trying to answer questions concerning the content of the video. (A)	0.844	3.09	2.11
It embarrasses me to fail answering content-based questions of the video. (A)	0.804	3.21	2.12
The performance demands coming from this video paralyze me. (A)	0.333	2.70	1.61
This video was a real challenge for me. (C) I am looking forward how well I have paid attention to this video. (C) I was fully intended on working hard on this video. (C) If I learn anything using this video, I am proud of my capability. (C).	No evaluation, as all four items charge onto different factors and therefore the construct is not reliable.		

The abbreviations behind each item stand for the affiliation with the corresponding factors. (I) stands for *interest*, (P) for *probability of success*, (A) for *anxiety*, and (C) for *challenge*.

Learning Gain

The last term we wanted to evaluate was *learning gain* which is traditionally evaluated by doing a pre-test before watching the video and doing a post-test after having watched the video. Both tests consist of the same questions as we want a reliable way of determining the learning gain. In total, there are 14 questions summing up in a total of 20 points. The learning gain of one attendee is determined by subtracting the achieved points of the pre-test from the achieved points of the post-test.

5.1.4. Presentation of the evaluation method

As we wanted to evaluate the differences between the two types of lecture recording concerning the terms fascination/interest, motivation and learning gain, we used the t-test for independent samples as, according to (Pospeschill, 2006) on page 213, it measures the difference of averages of two populations which are given by the two types of video.

Furthermore, (Pospeschill, 2006) says on page 213 that “*In order to apply and to interpret the t-statistics in an adequate way, it is a prerequisite that the datasets fulfill the assumption of the variance of homogeneity. According to this assumption, both populations have to possess equal variances. It is possible to prove this prerequisite by applying the F-test or the Levene test. If these tests show that the variances are not equal but in the case of a non-significant result, the assumption of the variance of homogeneity can still be retained according to the null hypothesis.*” Thus, we can apply this evaluation method.

5.2. Evaluation Results

After having done the surveys, we started to evaluate the data. As mentioned in the sections before, we applied a t-test to the data. Its independent variable is the type of the video, having two possibilities: *standard lecture recording* with one fixed video source and *enhanced lecture recording* with four video sources. The dependent variables are: *attentiveness*, *interest in the video*, three aspects of the motivation scale based on (Rheinberg, Vollmeyer & Burns, 2001), namely *QCM interest*, *QCM probability of success*, and *QCM anxiety*, and finally *learning gain*.

As mentioned above, we had 33 participants in our evaluation study, distributed over the two types of lecture recording. The standard lecture recording was watched by 17 attendees while the enhanced lecture recording was watched by 16.

The following tables show the results of the t-tests for every one of the six aspects we evaluated. In addition, we present the average and the standard deviation as charts separated by their underlying scales. While the aspects *attentiveness* and *interest in the video* are based on a numeric scale from 1 to 4, all *QCM* aspects are based on a numeric scale from 1 to 7, and the *learning gain* is based on a numeric scale from 0 to 20. Therefore, we present them in three different figures.

Table 15: t-test results for the construct "attentiveness".

Condition	Average	Std. Dev.	t-test	P
enhanced recording	2.24	0.713	-0.920	not significant
standard recording	2.46	0.670		

⇒ No significant differences between both types of lecture recording.

Table 16: t-test results for the construct "interest in the video".

Condition	Average	Std. Dev.	t-test	P
enhanced recording	2.09	0.957	0.254	not significant
standard recording	2.01	0.831		

⇒ No significant differences between both types of lecture recording.

When looking at Figure 39 both aspects reflecting the interest and the fascination in the videos are rated almost equally, and both have a high standard deviation. While the average of the attentiveness of the enhanced video is a little lower than the one of the standard video, the average value of the enhanced lecture recording concerning the interest in the video is slightly higher than the value of the standard lecture recording video.

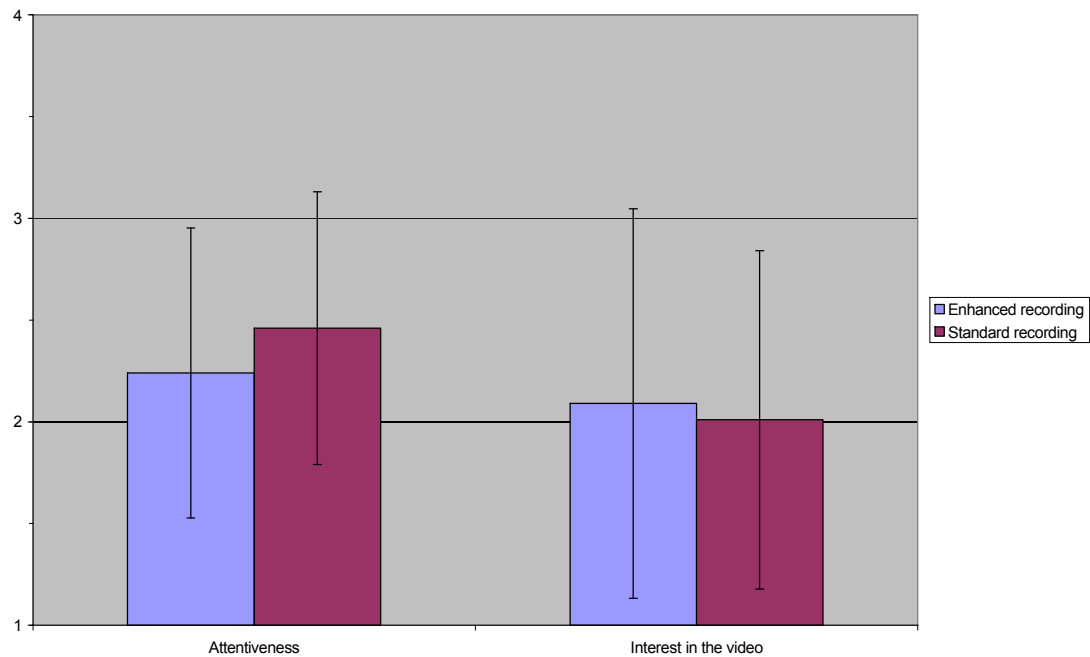


Figure 39: Average values of attentiveness and interest in the video compared.

QCM interest

Table 17: t-test results for the construct "QCM interest".

Condition	Average	Std. Dev.	t-test	<i>P</i>
enhanced recording	2.95	1.152	-0.641	not significant
standard recording	3.25	1.476		

⇒ No significant differences between both types of lecture recording.

QCM probability of success

Table 18: t-test results for the construct "QCM probability of success".

Condition	Average	Std. Dev.	t-test	<i>P</i>
enhanced recording	3.95	1.69	-0.435	not significant
standard recording	4.19	1.45		

⇒ No significant differences between both types of lecture recording.

QCM anxiety

Table 19: t-test results for the construct "QCM anxiety".

Condition	Average	Std. Dev.	t-test	<i>P</i>
enhanced recording	3.31	1.501	0.359	not significant
standard recording	3.10	1.541		

⇒ No significant differences between both types of lecture recording.

The results in Figure 40 show almost the same information as the figure before. Concerning the aspects of QCM interest and QCM probability of success, the enhanced lecture recording video is rated to be on the average slightly below the average value of the standard lecture recording video. Consequently, concerning the aspect of QCM anxiety, the enhanced video average value is slightly higher than the one for the standard video.

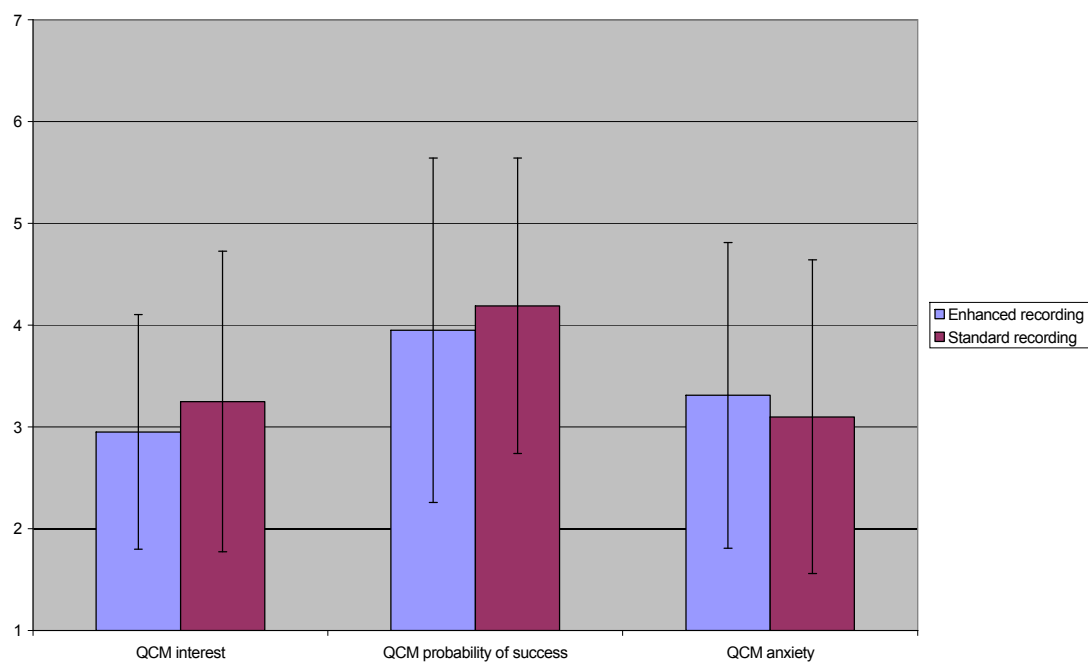


Figure 40: Average values of three QCM aspects compared.

Learning gain

Table 20: t-test results for the construct "learning gain".

Condition	Average	Std. Dev.	t-test	P
enhanced recording	9.81	3.48	0.584	not significant
standard recording	9.12	3.36		

⇒ No significant differences between both types of lecture recording.

Again, there is not much difference in the average learning gain between both types of lecture recording. The figure shows a slightly higher average learning gain when watching the enhanced recording of our Automatic Lecture Recording system.

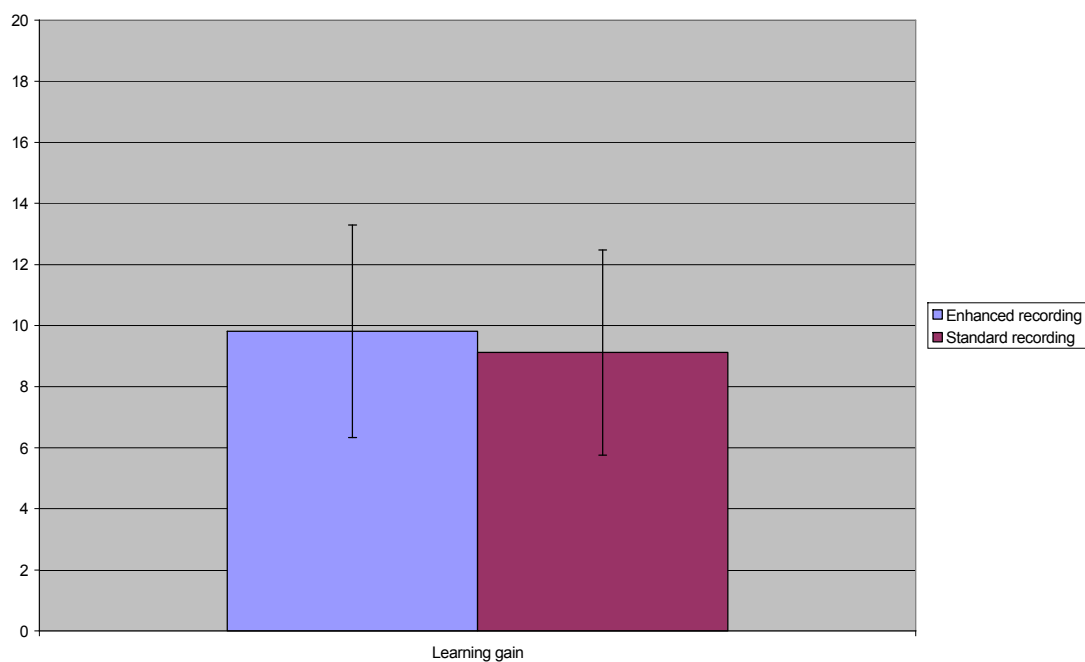


Figure 41: Average values of learning gain compared.

We amend this result by some more aspects concerning the motivation and the self-assessment of the attendees. At first, let us take a closer look at the motivation of the attendees. Are they unconcerned or are they sufficiently motivated? In our questionnaire, we asked five questions having a scale from 1 (*does not apply at all*) to 4 (*fully applies*). Table 21 shows the results.

Table 21: Statistic results of the items concerning the motivation.

<i>I watched this video because ...</i>	Average	Standard Deviation
... I have been payed for attending.	3.48	0.906
... I wanted to support the research in this area by attending.	3.12	0.893
... its content is important for my future career.	1.18	0.465
... I am interested in the technology presenting the learning matter.	1.76	0.902
... I like learning multimedia-based.	2.15	1.034

The type of lecture recording is insignificant for the motivation of the attendees. Therefore, we evaluated the items at once for all 33 participants. They all were extrinsically and intrinsically motivated, as the first two items show. In contrast, neither the technology nor the content shown was relevant to the participants. Even the relatively new technique of “multimedia-based learning” is only a moderate reason for attending the evaluation study.

Furthermore, it is of interest how the attendees rate their own ability of learning using such a medium. We evaluated the difference between the two types of lecture recordings by two items, the first having a scale from 1 (*one time*), 2 (*1 to 2 times*), 3 (*4 to 5 times*) to 4 (*more than 5 times*) while the second has a scale from 1 (*nothing*), 2 (*few*), 3 (*something*), to 4 (*very much*). The results are shown in the Tables 22 and 23.

Table 22: Assumed number of repetitions of lecture recordings.

	Average	Standard Deviation
How many times do you think you have to watch this video in order to perceive the entire learning matter of the lecture?	Enhanced: 2.44 Standard: 2.29	Enhanced: 0.727 Standard: 0.686

It is interesting that the attendees thought they would need slightly more repetitions of the lecture recordings of the enhanced video than of the standard video although they already had a slightly higher learning gain after having watched the video once. The self-assessment of the attendees before the post-test is shown in Table 23.

Table 23: Assumed learning gain.

	Average	Standard Deviation
According to your opinion, how much did you learn from this lecture by this kind of video direction?	Enhanced: 2.63 Standard: 2.53	Enhanced: 0.719 Standard: 0.514

5.3. Discussion of the evaluation

Unfortunately for us, all aspects we evaluated show no significant differences between the two types of lecture recordings. This is not the result we expected.

Besides the fact that we found no significant difference between the two types of lecture recordings, nearly all aspects we evaluated suggest that the enhanced recording performed slightly worse. There was only one exception that the enhanced lecture recording video performed slightly better concerning the aspect of learning gain which is very surprising but reassuring. Even though the attendees seemed to be uncertain how well they performed in learning from lecture recordings, especially from the enhanced version, they achieved a slightly higher learning gain on the average.

Furthermore, it is a good sign that the self-assessment of the attendees and the results in the learning gain have a tendency towards the enhanced lecture recording video.

From the evaluation we found three issues which could be improved in future work. At first, during the recording of our videos, two failures occurred: a) After about 20 minutes, the projector in the lecture hall failed, and we had to restart it in order to continue with our lecture. b) When showing the relevant example videos in the lecture, the lecturer forgot to tick the “slides only” switch, resulting in the virtual camera team in behaving as normal and even repeatedly switching away from the relevant slides camera. Thus it was only an issue for the enhanced lecture recording video as the standard video only shows the slides directly recorded from the VGA output of the lecturer’s computer.

The second issue is that in our study only students from study courses out of the social sciences took part who are not used to learn from lecture recordings. In order to improve the expressiveness of our evaluation study, students from many different study courses and faculties should take part.

The third issue is that we were only able to take a snapshot impression of the learning gain, mainly based on the short-term memory of the students. Thus, evaluating the aspects over a longer term could lead to more meaningful results.

Hence, preparing and running one or even better a number of evaluation studies, taking these issues into account, should lead to more significant results when comparing standard lecture recordings with the enhanced lecture recording videos of our distributed virtual camera team.

From my personal point of view, there are two more hypotheses in which our test setup differs from the real life and may lead to a different result. First, we conducted the evaluation inside the lecture hall which still gives a “community feeling” in contrast to the typical lonely learner at home while preparing for the exams. Second, there is a difference of the attitude towards learning from media if the result of the oncoming test is relevant for the students’ vita or not. It is obvious, that besides our time constraints it is impossible pushing students to take such a high risk for their career. Maybe, future work will give a solution.

6. Summary

In this dissertation we have presented the design and implementation of a distributed virtual camera team for Automatic Lecture Recording. At first, we checked the state-of-the-art of lecture recording, evaluated the related work, and investigated feasibility constraints. In the second step, human camera teams acted as models for our virtual team: we analyzed the work of the different camera team members in detail and extracted the requirements for their virtual equivalents. During the subsequent implementation phase, we were able to realize the requirements as well as to discover some drawbacks for which work-arounds were found and described in the experiences chapter. Finally, we conducted an evaluation of the system: its design, results and consequences were shown in Chapter five.

In the next sections, we conclude our findings before we give an outlook on how the vantages of the system evolve, and on possible future work.

6.1. The Virtual Camera Team

It is natural to use a real camera team as a role model for our system. Moreover, it is a good approach as the fragmentation of work is a basic principle to realize complex tasks and as the necessary team members can easily be described by comparing the necessities of Automatic Lecture Recording with the jobs the team members accomplish in reality. By determining the roles and the tasks of each role, it became clear which job had to be done by which module.

The main difference of our approach to similar work is that we have implemented several levels of *cinematographic rules* as it is the groundwork for each human camera team. In order to do so, we have implemented an extended finite state machine including contexts and conditions to enable the virtual director module to take its decisions in a diversified manner, always similar but seldom identical as well as not predictable but still comprehensible at run-time. In addition, image processing algorithms were implemented in the virtual cameraman module in order to enable its basic level of cinematographic rules by making autonomous decisions concerning the framing of shots as well as giving feedback to the virtual director module.

Establishing a communication channel between the virtual director module and the virtual cameraman modules also is borrowed from the human original in which a so called *Intercom* is used. Such a communication channel opens up the realization of a higher level of cinematographic rules which are more complex, e.g., the shot – counter-shot scenario during dialogs.

Finally, as cinematographic rules describe not only technical behaviors but also recommended reactions on events in reality we amended our system by sensor tools which add information about the environment. So, we enable the virtual camera team to properly react instead of randomly decide without any knowledge of the environment.

6.2. Implementation Experiences

Summing up Chapter 4, we state for the virtual director module that the contexts and the conditions are a robust working way to implement basic cinematographic rules. Its communication channels with the virtual cameramen and with the sensor tools module make the virtual director the central decision instance, which it is also in reality. As its configuration files are not hard-coded and thus exchangeable, the virtual director may be used for more than its original purpose as long as a new purpose can be mapped to such an extended finite state machine. It has proved to be fast and reliable over months having more and more complex cinematographic rules implemented compared to simpler-structured finite state machines.

For the virtual cameraman module, we conclude that the algorithms we use work sufficiently well and, even more important, in real-time. Therefore, the virtual cameraman module is able to realize basic cinematographic rules autonomously and to execute more sophisticated cinematographic rules in collaboration with the virtual director module.

The sensor tools module proved to be an excellent complement for the virtual director module as well as for the virtual cameraman module. It provides the necessary wits and knowledge to localize students in the audience and to interpret their announcement gestures, and it transports their spoken words. In addition, the QM software suite transfers the question and answer interaction of a lecture into the virtual world of our Automatic Lecture Recording system. We have mapped each involved role from the

real world into its digital equivalent and implemented all the necessary tasks. Our distributed system provides support for the complete question and answer dialog and aggregates all events in order to derive suitable sensor inputs for the director module of the Automatic Lecture Recording system.

The virtual sound engineer was implemented partially in the QM software suite of the sensor tools module and partially in the AV Mixer/Recorder. It turned out to be a feasible solution to process the audio data exactly at those points where the voice information is generated. For example, using the PDAs to sample the audio and receive the data of all PDAs at the QM server before feeding it into the audio part of the audience video server is feasible. Moreover, doing the final mixing and mastering in the last processing instance, the AV Mixer/Recorder, just before joining the audio and video streams is the common solution, the way it is done by broadcasting stations.

Finally, the AV Mixer/Recorder is able to process all orders from the virtual director module in real-time and to save the results on disk. It turned out that a high-end machine is necessary to cope with the requirements and numbers of uncompressed audio and video streams in combination with the audio and video effects applied (e.g., PiP).

6.3. Evaluation Experience

The main impulse for this dissertation was to increase the motivation of students watching the recorded videos and to provide students with an improved medium which some may prefer to learn from. However, the main goal for students is to improve the learning gain, which is not directly correlated to the idea of making it more convenient to watch the recording.

The results of our evaluation show that the videos recorded by our prototype of the distributed Automatic Lecture Recording system are slightly better accepted by the students than standard lecture recordings, especially when looking at the performance when learning from them. Referring to the section on related work, we state that standard lecture recordings as well as our approach are both successful in providing such a novel kind of learning media.

Furthermore, we showed three measures in order to a) improve the effect of the enhanced lecture recording video, b) put the evaluation on a wider population by integrating more students of a wider variety of study courses, and c) decrease incidental

influences by conducting the study over a longer term. Altogether, these measures can lead to results different from ours. Due to our time constraints, it was not possible to conduct such evaluation studies which therefore are important tasks for future work.

Another important task is to change the final video creation from the hard-disk approach, sufficient for recording, to a real-time approach necessary for live streaming.

6.4. Rating the Prototype

In order to rate our approach of the distributed Automatic Lecture Recording software system, we look back to the constraints, limitations, and prerequisites listed in the beginning of this paper. We were concerned about aesthetic and financial constraints as well as about limitations in space and time. So, let us now go through these requirements and rate our approach.

6.4.1. Aesthetic Approach

In order to meet the demands of an aesthetic approach, we took the work of a human camera team as our role model. Our focus was not only to set up a distributed system where its modules bear the names of the different team members, but to enable these modules to mimic the work of their comparable human pendants.

We were successful in implementing basic cinematographic rules as well as more complex ones, namely

- respect the “line of action” while using four cameras,
- ensuring the minimal and recommended duration of a shot to fit the type of recording,
- avoiding to show a specific shot too often,
- avoiding to show recurring and therefore predictable sequences of shots in order to keep the spectator more focused on the content,
- making decisions which shot to show next, based on events in the environment registered by various sensors,
- Quick responses to events,

- inserting *neutral shots* to keep the spectator informed about the environment of the lecture,
- autonomous framing and adjusting of exposure parameters depending on the current scene,
- autonomous reaction on gesticulating and/or moving protagonists,
- arranging and conducting of shot – counter-shot scenarios,
- use of transition effects and picture-in-picture effects for the video in order to react in a meaningful way on a larger number of incoming events from different sensors.

As we were able to realize quite a few of the relevant cinematographic rules, we conclude that the aesthetic claim has been satisfied. The implementation of further cinematographic rules by future work will continuously improve our result so far.

6.4.2. Affordable Approach

One important requirement is to minimize the costs to set up an Automatic Lecture Recording system in the lecture hall. In contrast to the huge expenditures presented in the Chapters 1.3.3 and 2.1.3, the cost for our prototype is mainly a one-time investment into the equipment as the entire system needs at most one operator for starting and stopping if it is not done by the lecturer him- or herself.

Comparing the one-time cost of our system of about 11,200 € for the equipment with no extra costs for staff to the investment into professional AV recording equipment of more than 20,000 € and recurring costs of about 1,700 € for the crew per recording day, or, as another possibility, even compared to the recurring costs for renting the equipment and the crew of about 6,500 € per recording day, it is obvious that our approach is much more affordable.

6.4.3. Space- and Time-Saving Approach

As our approach is based on AV streams over the network and on controlling the equipment over this network, it is no problem to place the computers in a different room than the lecture hall. While the cameras can be mounted fixed in the lecture hall, it is useful to keep the video server for the slides attached to the presentation computer

of the lecturer. Therefore, space is no issue as all necessary devices can be spread over different rooms.

Besides being frugal concerning space consumption, our approach helps to save the lecturer's time and even that of operating staff. The entire system works autonomously after being started manually until it is stopped manually. Thus, only at the beginning and at the end of a lecture, human action is necessary. Furthermore, as we also use our AV Mixer/Recorder computer to transcode the resulting lecture recording into many different streaming formats, and as we have automated this transcoding process and the publication on a streaming server (Lampi, Kopf & Effelsberg, 2006), the entire chain of providing students with lecture recordings is fast. Of course, it is still possible to manually perform any post-production steps (e.g., removing errors the lecturer made in class) before starting the transcoding.

Even if it is not possible to mount the equipment in a fixed way in the lecture hall, it is not very time-consuming to set it up. For our prototype, we brought all the equipment consisting of the computers for the lecturer's presentation, the director computer, the cameramen computer, three PTZ cameras, the video server, the scan converter and all the necessary cabling into the lecture hall every time we recorded a lecture which was at least twice a week during the term. It was no problem to set the system up during the normal break between two lectures, which is 15 minutes, as we used a trolley on which most of the equipment was kept. After getting used to it, it took only about five minutes to set the system up from scratch.

6.4.4. Successful Prototype

Concluding the rating of our prototype, we assert that the initial requirements are all fulfilled. Therefore, we state that our approach is successful. When looking at our evaluation results and the evaluation results of (Rui, Gupta, Grudin, 2003) it seems to be realistic that an automated mimicry of a real camera team can still be detected even by non-professionals. In both cases humans detected the small things that differ to a real camera team and judged accordingly. Nevertheless, for all cases of lecture recording in which it is not feasible to employ a human camera team, our prototype can be a successful substitute. Furthermore, it is promising as a good base for future work which we will describe in the next Sections.

6.5. Outlook

From our current point of view, there are three important ways how to continue our work:

1. working on improvements of the current implementation concerning, e.g., faster or more precise algorithms especially in image and video processing, or more intuitive user interfaces;
2. working on new implementations which add new features to our prototype;
3. transferring our approach to a wider set of applications.

The following Sections discuss the details of these three ways.

6.5.1. Improving the Current Prototype

Image processing algorithms naturally are a good area to improve performance and precision. In order to improve a shot – counter-shot scenario, the implementation of face detection, of gaze direction detection, and of visual person localization would be very helpful. Even more, any algorithm able to extract semantic meaning out of image processing will improve the system, e.g., reliable handraising detection. Furthermore, audio processing algorithms are able to support not only the audio quality but also the location of persons.

This leads directly to an important point: Up to now the question announcement detection is based on a GUI on a PDA, and hitting a button on a PDA really is a non-intuitive way to announce a question. Therefore, research on how to detect a hand-raising questioner in the audience and determine his or her position by using video and audio processing techniques has just been done by (Herweh, 2009).

Another internal improvement is to implement more cinematographic rules. This requires additional sensors and investigations how to interpret their signals and measurements, similar to the way a human camera team would interpret them.

Improvements concerning the usability are twofold: from the point of view of the system operator, it would be useful to implement an automated sequence to start up the different software parts of the system instead of doing it manually. It would also be nice to improve the automation of the transcoding system.

From the lecturer's point of view, the GUI can be improved, e.g., by giving the lecturer the detailed control when to block questions and when to limit the shots to be recorded to those including the slides camera, e.g., when the lecturer shows animations, simulations, or videos. Furthermore, showing the position of an announcing questioner graphically on the GUI will help the lecturer to get into eye contact with the questioner. It would also be nice to provide improved support for questioners in remote lecture halls.

From the questioner's point of view, implementing the announcement detection in a more intuitive way and porting the QM client software from PDAs to notebooks would be useful improvements. Even though it is necessary to port the software to different operating systems it is useful as most students already use a notebook or a netbook during the lecture. Problems with insufficient batteries will disappear, and the quality of audio transmitted data over WLAN will increase as notebooks and netbooks usually have better equipment built in. The most important operating systems to port the QM questioner software to are Microsoft Windows, Apple Mac OS and Linux. As the software solves relatively simple tasks, and the protocols used are simple, too, porting can be done e.g., by students during a student project.

Finally, concerning the usability, it could be useful to conduct empirical evaluations in order to investigate the influence of the system on a lecture, e.g., to which extent a lecturer and the questioners are distracted by the system.

6.5.2. Extending the Current Prototype

Extending the current prototype in order to enable it for *live* streaming is based on two parts which we presented in Chapter 4.5.3. As already mentioned, both parts are obvious but due to the given time constraints it was not possible to fully implement them, respectively to test them completely.

The first part is a replacement for the audio normalizing algorithm with combined algorithms of a noise gate, an expander, a compressor and a limiter. The main advantage is that this combination does not need to detect a global maximum before being applied.

The second part, necessary to enable live streaming, are “DirectShow Source Filters” accepting bitmaps respectively raw PCM data as their input, and “DirectShow Transform Filters” transmitting the encoded streams using the RTP protocol.

6.5.3. Transferring Automatic Lecture Recording to other environments

As our prototype of the distributed Automatic Lecture Recording system is already successful in its first version, it is worth thinking of transferring it to different contexts outside lectures. From a technical point of view, all necessary parts are configurable so that there are no principle restrictions. Mainly, the FSM has to be re-written to cover all possible situations of the new context, and the configuration files have to be adapted to the hall where the event takes place.

Nevertheless, there currently are three constraints:

1. If the event to be covered by the FSM is very complex, it is really hard to manually write an FSM covering all possible situations. There might be a limit to the events which can be covered by our Automatic Recording system. The complexity at which such a limit occurs should be examined in future work.
2. Up to now, only fixed, mounted PTZ cameras have been used. If autonomous, moving cameras with any degrees of freedom are getting employed, many of the algorithms of the virtual cameraman module have to be rewritten as new kinds of motion will appear in the image.
3. Spontaneous recordings, e.g., on the street, are not the target of our Automatic Lecture Recording system as there is significant effort to be done to measure and to calibrate the system to every new location.

However, it should be easy to adapt the Automatic Lecture Recording system to all kinds of frontal presentations as this genre has rigid rules of interaction. The occurrences of this genre are manifold, e.g., internal presentations in companies, presentation coaching, panel discussions, plenary meetings, party conventions, stockholders’ meetings and court hearings.

Transferring the system may also mean to exchange the recording equipment to meet different technical requirements, e.g., to increase the quality. As the cameraman is built modularly it is quite easy to exchange the cameras as long as they provide the

same set of functionals. If the functional range is different it is necessary to rewrite parts of the virtual cameraman module.

Depending on the equipment in use, higher system requirements may arise. To be more precise, while the Axis cameras and video servers used to consume a total bandwidth of about 16 Mbit/s during run-time, one single professional camera using as its main output the serial digital interface (SDI) has a constant bit rate of 270 Mbit/s for standard definition (SD) resolution, defined by the SMPTE-259M standard. In case of high definition (HD) resolution cameras, the main output uses the so-called HD-SDI with a constant bit rate of 1.485 Gbit/s, according to the SMPTE-292M standard. In near future the new 3 Gbit/s standard will be common in the studios. It is obvious that these amounts of data require much higher system capacities for every part of the system in order to keep up with the real-time requirement.

Having the already implemented features in mind and aiming at the presented possibilities of future work, this project is not only able to bring Automatic Lecture Recording to a higher level but also to extend its scope to further applications, providing a wider basis for researchers.

7. Appendix

7.1. Configuration Files

7.1.1. XML file of the FSM used in our prototype

```
<?xml version="1.0" encoding="UTF-8" ?>
<FSM>
  <Name>Lecture</Name>
  <Version>0.2 Draft</Version>
  <Description>Basic Lecture Recording Draft</Description>
  <Author>Fleming Lampi</Author>
  <Date>y2006.m04.d05</Date>
  <Metadata>
    <FirstTag>First value</FirstTag>
    <SecondTag>Second value</SecondTag>
  </Metadata>

  <Director>
    <IP>134.155.92.68</IP>
  </Director>

  <Cameras>
    <Camera>
      <Name>Audience</Name>
      <RTSP>rtsp://134.155.92.47:1026/mpeg4/1/media.amp</RTSP>
    </Camera>
    <Camera>
      <Name>Lecturer</Name>
      <RTSP>rtsp://134.155.92.23:1024/mpeg4/1/media.amp</RTSP>
    </Camera>
    <Camera>
      <Name>Slides</Name>
      <RTSP>rtsp://134.155.92.80:1027/mpeg4/1/media.amp</RTSP>
    </Camera>
    <Camera>
      <Name>LongShot</Name>
      <RTSP>rtsp://134.155.92.74:1025/mpeg4/1/media.amp</RTSP>
    </Camera>
  </Cameras>

  <Recorder>
    <IP>134.155.92.12</IP>
    <Port>49901</Port>
  </Recorder>

  <Contexts>
    <Context>
      <Number>0</Number>
      <Name>out of context</Name>
    </Context>
    <Context>
      <Number>1</Number>
      <Name>lecture context</Name>
    </Context>
    <Context>
      <Number>2</Number>
      <Name>question context</Name>
    </Context>
    <Context>
      <Number>3</Number>
      <Name>answer context</Name>
    </Context>
  </Contexts>

  <CuttingTypes>
    <Type>
      <Number>1</Number>
      <Name>Cut</Name> <!-- Schnitt -->
      <SourceChange>Yes</SourceChange>
    </Type>
    <Type>
      <Number>2</Number>
      <Name>Fade</Name> <!-- Blende -->
      <SourceChange>Yes</SourceChange>
    </Type>
    <Type>
      <Number>3</Number>
      <Name>Pan/Tilt</Name> <!-- Schwenken/Neigen (Object wechseln) -->
      <SourceChange>Yes</SourceChange>
    </Type>
    <Type>
      <Number>4</Number>
      <Name>Pan/Tilt</Name> <!-- Schwenken/Neigen (Object beibehalten) -->
      <SourceChange>No</SourceChange>
    </Type>
    <Type>
      <Number>5</Number>
      <Name>Zoom in</Name> <!-- Zufahrt -->
      <SourceChange>No</SourceChange>
    </Type>
  </CuttingTypes>

```

```

    <Number>6</Number>
    <Name>Zoom out</Name> <!-- Aufzieher -->
    <SourceChange>No</SourceChange>
  </Type>
</Type>
  <Number>7</Number>
  <Name>Frame left</Name> <!-- links ins Bild setzen -->
  <SourceChange>No</SourceChange>
</Type>
</Type>
  <Number>8</Number>
  <Name>Frame mid</Name> <!-- mittig ins Bild setzen -->
  <SourceChange>No</SourceChange>
</Type>
</Type>
  <Number>9</Number>
  <Name>Frame right</Name> <!-- rechts ins Bild setzen -->
  <SourceChange>No</SourceChange>
</Type>
</CuttingTypes>

<ConditionTypes>
  <CondType>
    <Number>0</Number>
    <Name>time</Name>
  </CondType>
  <CondType>
    <Number>-1</Number>
    <Name>still</Name>
  </CondType>
  <CondType>
    <Number>1</Number>
    <Name>gesticulating</Name>
  </CondType>
  <CondType>
    <Number>-2</Number>
    <Name>notMoving</Name>
  </CondType>
  <CondType>
    <Number>2</Number>
    <Name>moving</Name>
  </CondType>
  <CondType>
    <Number>-3</Number>
    <Name>inactive</Name>
  </CondType>
  <CondType>
    <Number>3</Number>
    <Name>active</Name>
  </CondType>
  <CondType>
    <Number>-4</Number>
    <Name>noSpace</Name>
  </CondType>
  <CondType>
    <Number>4</Number>
    <Name>space</Name>
  </CondType>
  <CondType>
    <Number>-5</Number>
    <Name>denied</Name>
  </CondType>
  <CondType>
    <Number>5</Number>
    <Name>acknowledged</Name>
  </CondType>
  <CondType>
    <Number>-6</Number>
    <Name>answerFalse</Name>
  </CondType>
  <CondType>
    <Number>6</Number>
    <Name>answerOK</Name>
  </CondType>
  <CondType>
    <Number>-7</Number>
    <Name>quiet</Name>
  </CondType>
  <CondType>
    <Number>7</Number>
    <Name>speaking</Name>
  </CondType>
  <CondType>
    <Number>-8</Number>
    <Name>annotate</Name>
  </CondType>
  <CondType>
    <Number>8</Number>
    <Name>switch</Name>
  </CondType>
  <CondType>
    <Number>-9</Number>
    <Name>stop</Name>
  </CondType>
  <CondType>
    <Number>9</Number>
    <Name>deferred</Name>
  </CondType>
  <CondType>
    <Number>10</Number>
    <Name>answering</Name>
  </CondType>

```

```

    </CondType>
  </ConditionTypes>

  <ConditionObjects>
    <CondObject>
      <Number>1</Number>
      <Name>lecturer</Name>
    </CondObject>
    <CondObject>
      <Number>2</Number>
      <Name>questioner</Name>
    </CondObject>
    <CondObject>
      <Number>3</Number>
      <Name>audience</Name>
    </CondObject>
    <CondObject>
      <Number>4</Number>
      <Name>slide</Name>
    </CondObject>
    <CondObject>
      <Number>5</Number>
      <Name>event</Name>
    </CondObject>
  </ConditionObjects>

  <Definition>
    <Startstate>1</Startstate>

    <State>
      <Number>1</Number>
      <Name>Start</Name>
      <Context>0</Context>
      <Camera>LongShot</Camera>
      <CameraPiP>Empty</CameraPiP>
      <Transitions>
        <Event>
          <Possibilities>
            <Possibility>
              <Number>1</Number>
              <Type>StartOfLecture</Type>
              <NewState>2</NewState>
              <CutType>1</CutType>
              <Conditions>Empty</Conditions>
            </Possibility>
          </Possibilities>
        </Event>
      </Transitions>
    </State>

    <State>
      <Number>2</Number>
      <Name>Very Long Shot Lecturer</Name>
      <Context>1</Context>
      <Camera>LongShot</Camera>
      <CameraPiP>Empty</CameraPiP>
      <Transitions>
        <Time>
          <min>15</min>
          <recommend>18</recommend>
          <max>30</max>
          <randomRange>20%</randomRange>
        </Time>
        <Possibilities>
          <Possibility>
            <Number>1</Number>
            <Type>Time</Type>
            <NewState>3</NewState>
            <CutType>1,2</CutType>
            <Conditions>
              <lecturer>moving,active, speaking</lecturer>
            </Conditions>
          </Possibility>
          <Possibility>
            <Number>2</Number>
            <Type>Time</Type>
            <NewState>2</NewState>
            <CutType>5</CutType>
            <Conditions>
              <lecturer>still keeping,active</lecturer>
            </Conditions>
          </Possibility>
          <Possibility>
            <Number>3</Number>
            <Type>Time</Type>
            <NewState>5</NewState>
            <CutType>1,2,4</CutType>
            <Conditions>
              <slide>active,space,switch,annotate</slide>
              <lecturer>active</lecturer>
            </Conditions>
          </Possibility>
          <Possibility>
            <Number>4</Number>
            <Type>Time</Type>
            <NewState>6</NewState>
            <CutType>1,2</CutType>
            <Conditions>
              <slide>active,nospace,switch,annotate</slide>
            </Conditions>
          </Possibility>
        </Possibilities>
      </Transitions>
    </State>
  </Definition>

```

```

</Time>
<Event>
  <Possibilities>
    <Possibility>
      <Number>1</Number>
      <Type>End Of Lecture</Type>
      <NewState>15</NewState>
      <CutType>1</CutType>
      <Conditions>Empty</Conditions>
    </Possibility>
    <Possibility>
      <Number>2</Number>
      <Type>Question Acknowledged</Type>
      <NewState>7</NewState>
      <CutType>1</CutType>
      <Conditions>
        <questioner>acknowledged</questioner>
      </Conditions>
    </Possibility>
  </Possibilities>
</Event>
</Transitions>
</State>

<State>
  <Number>3</Number>
  <Name>Medium Shot Lecturer</Name>
  <Context>1</Context>
  <Camera>Lecturer</Camera>
  <CameraPiP>Empty</CameraPiP>
  <Transitions>
    <Time>
      <min>15</min>
      <recommend>20</recommend>
      <max>30</max>
      <randomRange>20%</randomRange>
    </Time>
    <Possibilities>
      <Possibility>
        <Number>1</Number>
        <Type>Time</Type>
        <NewState>2</NewState>
        <CutType>1,6</CutType>
        <Conditions>
          <slide>inactive</slide>
          <lecturer>inactive</lecturer>
        </Conditions>
      </Possibility>
      <Possibility>
        <Number>2</Number>
        <Type>Time</Type>
        <NewState>4</NewState>
        <CutType>1</CutType>
        <Conditions>
          <lecturer>notMoving</lecturer>
          <audience>active</audience>
        </Conditions>
      </Possibility>
      <Possibility>
        <Number>3</Number>
        <Type>Time</Type>
        <NewState>5</NewState>
        <CutType>1,2,4</CutType>
        <Conditions>
          <slide>active,space,switch,annotate</slide>
          <lecturer>active</lecturer>
        </Conditions>
      </Possibility>
      <Possibility>
        <Number>4</Number>
        <Type>Time</Type>
        <NewState>6</NewState>
        <CutType>1,2</CutType>
        <Conditions>
          <slide>active,nospace,switch,annotate</slide>
        </Conditions>
      </Possibility>
      <Possibility>
        <Number>5</Number>
        <Type>Time</Type>
        <NewState>3</NewState>
        <CutType>4</CutType>
        <Conditions>
          <lecturer>still,active</lecturer>
        </Conditions>
      </Possibility>
      <Possibility>
        <Number>6</Number>
        <Type>Time</Type>
        <NewState>3</NewState>
        <CutType>6</CutType>
        <Conditions>
          <lecturer>gesticulating,active</lecturer>
        </Conditions>
      </Possibility>
    </Possibilities>
  </Time>
<Event>
  <Possibilities>
    <Possibility>
      <Number>1</Number>
      <Type>End Of Lecture</Type>

```



```

        <NewState>15</NewState>
        <CutType>1</CutType>
        <Conditions>Empty</Conditions>
    </Possibility>
    <Possibility>
        <Number>2</Number>
        <Type>Question Acknowledged</Type>
        <NewState>7</NewState>
        <CutType>1</CutType>
        <Conditions>
            <questioner>acknowledged</questioner>
        </Conditions>
    </Possibility>
    <!--<Possibility>
        <Number>3</Number>
        <Type>Slide Switch</Type>
        <NewState>6</NewState>
        <CutType>1</CutType>
        <Conditions>
            <slide>switch, annotate</slide>
        </Conditions>
    </Possibility>-->
    </Possibilities>
</Event>
</Transitions>
</State>

<State>
    <Number>4</Number>
    <Name>Medium to Long Shot Audience</Name>
    <Context>1</Context>
    <Camera>Audience</Camera>
    <CameraPiP>Empty</CameraPiP>
    <Transitions>
        <Time>
            <min>15</min>
            <recommend>18</recommend>
            <max>30</max>
            <randomRange>20%</randomRange>
        </Time>
        <Possibilities>
            <Possibility>
                <Number>1</Number>
                <Type>Time</Type>
                <NewState>3</NewState>
                <CutType>1, 2, 6</CutType>
                <Conditions>
                    <lecturer>moving, speaking, active</lecturer>
                    <slide>inactive</slide>
                </Conditions>
            </Possibility>
            <Possibility>
                <Number>2</Number>
                <Type>Time</Type>
                <NewState>6</NewState>
                <CutType>1, 2</CutType>
                <Conditions>
                    <slide>active, switch, annotate</slide>
                </Conditions>
            </Possibility>
            <Possibility>
                <Number>3</Number>
                <Type>Time</Type>
                <NewState>4</NewState>
                <CutType>5</CutType>
                <Conditions>
                    <audience>active</audience>
                </Conditions>
            </Possibility>
            <Possibility>
                <Number>4</Number>
                <Type>Time</Type>
                <NewState>4</NewState>
                <CutType>6</CutType>
                <Conditions>
                    <audience>active</audience>
                </Conditions>
            </Possibility>
            <Possibility>
                <Number>5</Number>
                <Type>Time</Type>
                <NewState>2</NewState>
                <CutType>1, 6</CutType>
                <Conditions>
                    <slide>inactive</slide>
                    <lecturer>inactive</lecturer>
                </Conditions>
            </Possibility>
        </Possibilities>
    </Time>
</Event>
    <Possibilities>
        <Possibility>
            <Number>1</Number>
            <Type>End Of Lecture</Type>
            <NewState>15</NewState>
            <CutType>1</CutType>
            <Conditions>Empty</Conditions>
        </Possibility>
        <Possibility>
            <Number>2</Number>
            <Type>Question Acknowledged</Type>

```

```

        <NewState>7</NewState>
        <CutType>1</CutType>
        <Conditions>
            <questioner>acknowledged</questioner>
        </Conditions>
    </Possibility>
    <!--<Possibility>
        <Number>3</Number>
        <Type>Slide Switch</Type>
        <NewState>6</NewState>
        <CutType>1</CutType>
        <Conditions>
            <slide>switch, annotate</slide>
        </Conditions>
    </Possibility>-->
</Possibilities>
</Event>
</Transitions>
</State>

<State>
    <Number>5</Number>
    <Name>Lecturer PiP + Slide</Name>
    <Context>1</Context>
    <Camera>Slides</Camera>
    <CameraPiP>Lecturer</CameraPiP>
    <Transitions>
        <Time>
            <min>20</min>
            <recommend>30</recommend>
            <max>40</max>
            <randomRange>20%</randomRange>
        <Possibilities>
            <Possibility>
                <Number>1</Number>
                <Type>Time</Type>
                <NewState>3</NewState>
                <CutType>1,2</CutType>
                <Conditions>
                    <lecturer>active</lecturer>
                    <slide>inactive</slide>
                </Conditions>
            </Possibility>
            <Possibility>
                <Number>2</Number>
                <Type>Time</Type>
                <NewState>6</NewState>
                <CutType>1,2</CutType>
                <Conditions>
                    <slide>active, nospace, switch, annotate</slide>
                </Conditions>
            </Possibility>
            <Possibility>
                <Number>3</Number>
                <Type>Time</Type>
                <NewState>5</NewState>
                <CutType>4</CutType>
                <Conditions>
                    <lecturer>active, still</lecturer>
                    <slide>active, space, switch, annotate</slide>
                </Conditions>
            </Possibility>
            <Possibility>
                <Number>4</Number>
                <Type>Time</Type>
                <NewState>5</NewState>
                <CutType>6</CutType>
                <Conditions>
                    <lecturer>active, gesticulating</lecturer>
                    <slide>active, space, switch, annotate</slide>
                </Conditions>
            </Possibility>
            <Possibility>
                <Number>5</Number>
                <Type>Time</Type>
                <NewState>2</NewState>
                <CutType>1,6</CutType>
                <Conditions>
                    <slide>inactive</slide>
                    <lecturer>inactive</lecturer>
                </Conditions>
            </Possibility>
        </Possibilities>
    </Time>
    <Event>
        <Possibilities>
            <Possibility>
                <Number>1</Number>
                <Type>End Of Lecture</Type>
                <NewState>15</NewState>
                <CutType>1</CutType>
                <Conditions>Empty</Conditions>
            </Possibility>
            <Possibility>
                <Number>2</Number>
                <Type>Question Acknowledged</Type>
                <NewState>7</NewState>
                <CutType>1</CutType>
                <Conditions>
                    <questioner>acknowledged</questioner>
                </Conditions>
            </Possibility>
        </Possibilities>
    </Event>

```

```

        </Possibility>
    </Possibilities>
</Event>
</Transitions>
</State>

<State>
    <Number>6</Number>
    <Name>Slide</Name>
    <Context>1</Context>
    <Camera>Slides</Camera>
    <CameraPiP>Empty</CameraPiP>
    <Transitions>
        <Time>
            <min>20</min>
            <recommend>30</recommend>
            <max>40</max>
            <randomRange>20%</randomRange>
        </Time>
        <Possibilities>
            <Possibility>
                <Number>1</Number>
                <Type>Time</Type>
                <NewState>3</NewState>
                <CutType>1,2</CutType>
                <Conditions>
                    <lecturer>active</lecturer>
                    <slide>inactive</slide>
                </Conditions>
            </Possibility>
            <Possibility>
                <Number>2</Number>
                <Type>Time</Type>
                <NewState>4</NewState>
                <CutType>1,2</CutType>
                <Conditions>
                    <audience>active</audience>
                    <slide>inactive</slide>
                </Conditions>
            </Possibility>
            <Possibility>
                <Number>3</Number>
                <Type>Time</Type>
                <NewState>5</NewState>
                <CutType>1,2</CutType>
                <Conditions>
                    <lecturer>active</lecturer>
                    <slide>active,nospace,annotate,switch</slide>
                </Conditions>
            </Possibility>
            <Possibility>
                <Number>4</Number>
                <Type>Time</Type>
                <NewState>2</NewState>
                <CutType>1,6</CutType>
                <Conditions>
                    <slide>inactive</slide>
                    <lecturer>inactive</lecturer>
                </Conditions>
            </Possibility>
        </Possibilities>
    </Time>
<Event>
    <Possibilities>
        <Possibility>
            <Number>1</Number>
            <Type>End Of Lecture</Type>
            <NewState>15</NewState>
            <CutType>1</CutType>
            <Conditions>Empty</Conditions>
        </Possibility>
        <Possibility>
            <Number>2</Number>
            <Type>Question Acknowledged</Type>
            <NewState>7</NewState>
            <CutType>1</CutType>
            <Conditions>
                <questioner>acknowledged</questioner>
            </Conditions>
        </Possibility>
        <!--<Possibility>
            <Number>3</Number>
            <Type>Slide NoSpace</Type>
            <NewState>5</NewState>
            <CutType>1</CutType>
            <Conditions>
                <slide>nospace</slide>
            </Conditions>
        </Possibility>-->
    </Possibilities>
</Event>
</Transitions>
</State>

<State>
    <Number>7</Number>
    <Name>Medium Shot Questioner</Name>
    <Context>2</Context>
    <Camera>Audience</Camera>
    <CameraPiP>Empty</CameraPiP>
    <Transitions>
        <Time>

```

```

<min>15</min>
<recommend>18</recommend>
<max>30</max>
<randomRange>20%</randomRange>
<Possibilities>
  <Possibility>
    <Number>1</Number>
    <Type>Time</Type>
    <NewState>8</NewState>
    <CutType>1,2</CutType>
    <Conditions>
      <lecturer>active</lecturer>
      <slide>inactive</slide>
      <questioner>active</questioner>
    </Conditions>
  </Possibility>
  <Possibility>
    <Number>2</Number>
    <Type>Time</Type>
    <NewState>9</NewState>
    <CutType>1,2</CutType>
    <Conditions>
      <slide>active,space,switch,annotate</slide>
      <questioner>active</questioner>
    </Conditions>
  </Possibility>
  <Possibility>
    <Number>3</Number>
    <Type>Time</Type>
    <NewState>10</NewState>
    <CutType>1,2</CutType>
    <Conditions>
      <slide>active,nospace,switch,annotate</slide>
      <questioner>active</questioner>
    </Conditions>
  </Possibility>
  <Possibility>
    <Number>4</Number>
    <Type>Time</Type>
    <NewState>7</NewState>
    <CutType>4</CutType>
    <Conditions>
      <questioner>active,still</questioner>
    </Conditions>
  </Possibility>
  <Possibility>
    <Number>5</Number>
    <Type>Time</Type>
    <NewState>7</NewState>
    <CutType>6</CutType>
    <Conditions>
      <questioner>active,gesticulating</questioner>
    </Conditions>
  </Possibility>
</Possibilities>
</Time>
<Event>
  <Possibilities>
    <Possibility>
      <Number>1</Number>
      <Type>Question Denied</Type>
      <NewState>3</NewState>
      <CutType>1</CutType>
      <Conditions>
        <questioner>denied,deferred</questioner>
      </Conditions>
    </Possibility>
    <Possibility>
      <Number>2</Number>
      <Type>End Of Lecture</Type>
      <NewState>15</NewState>
      <CutType>1</CutType>
      <Conditions>Empty</Conditions>
    </Possibility>
    <Possibility>
      <Number>3</Number>
      <Type>Lecturer Answering</Type>
      <NewState>11</NewState>
      <CutType>1</CutType>
      <Conditions>
        <lecturer>answering</lecturer>
      </Conditions>
    </Possibility>
    <Possibility>
      <Number>4</Number>
      <Type>Slide Switch</Type>
      <NewState>9</NewState>
      <CutType>1</CutType>
      <Conditions>
        <slide>switch,annotate,space</slide>
      </Conditions>
    </Possibility>
    <Possibility>
      <Number>5</Number>
      <Type>Answer OK</Type>
      <NewState>3</NewState>
      <CutType>1</CutType>
      <Conditions>
        <lecturer>AnswerOK</lecturer>
      </Conditions>
    </Possibility>
  </Possibilities>

```

```

        <!--<Possibility>
            <Number>6</Number>
            <Type>Slide NoSpace</Type>
            <NewState>10</NewState>
            <CutType>1</CutType>
            <Conditions>
                <slide>nospace</slide>
            </Conditions>
        </Possibility>-->
    </Possibilities>
</Event>
</Transitions>
</State>

<State>
    <Number>8</Number>
    <Name>Questioner PiP + Lecturer</Name>
    <Context>2</Context>
    <Camera>Lecturer</Camera>
    <CameraPiP>Audience</CameraPiP>
    <Transitions>
        <Time>
            <min>15</min>
            <recommend>20</recommend>
            <max>30</max>
            <randomRange>20%</randomRange>
        </Time>
        <Possibilities>
            <Possibility>
                <Number>1</Number>
                <Type>Time</Type>
                <NewState>7</NewState>
                <CutType>1,2</CutType>
                <Conditions>
                    <lecturer>inactive</lecturer>
                    <slide>inactive</slide>
                    <questioner>active</questioner>
                </Conditions>
            </Possibility>
            <Possibility>
                <Number>2</Number>
                <Type>Time</Type>
                <NewState>11</NewState>
                <CutType>1,2</CutType>
                <Conditions>
                    <lecturer>active</lecturer>
                    <questioner>inactive</questioner>
                </Conditions>
            </Possibility>
            <Possibility>
                <Number>3</Number>
                <Type>Time</Type>
                <NewState>8</NewState>
                <CutType>5</CutType>
                <Conditions>
                    <lecturer>active,still</lecturer>
                    <questioner>active,still</questioner>
                </Conditions>
            </Possibility>
            <Possibility>
                <Number>4</Number>
                <Type>Time</Type>
                <NewState>8</NewState>
                <CutType>6</CutType>
                <Conditions>
                    <lecturer>active,gesticulating</lecturer>
                    <questioner>active,gesticulating</questioner>
                </Conditions>
            </Possibility>
        </Possibilities>
    </Time>
</Event>
<Possibilities>
    <Possibility>
        <Number>1</Number>
        <Type>Question Stopped</Type>
        <NewState>3</NewState>
        <CutType>1</CutType>
        <Conditions>
            <questioner>denied,deferred</questioner>
        </Conditions>
    </Possibility>
    <Possibility>
        <Number>2</Number>
        <Type>Slide Annotate</Type>
        <NewState>9</NewState>
        <CutType>1</CutType>
        <Conditions>
            <slide>annotate,space</slide>
        </Conditions>
    </Possibility>
    <Possibility>
        <Number>3</Number>
        <Type>Slide Annotate</Type>
        <NewState>10</NewState>
        <CutType>1</CutType>
        <Conditions>
            <slide>annotate,nospace</slide>
        </Conditions>
    </Possibility>
    <Possibility>
        <Number>4</Number>

```

```

        <Type>End Of Lecture</Type>
        <NewState>15</NewState>
        <CutType>1</CutType>
        <Conditions>Empty</Conditions>
    </Possibility>
    <Possibility>
        <Number>5</Number>
        <Type>Lecturer Answering</Type>
        <NewState>12</NewState>
        <CutType>1</CutType>
        <Conditions>
            <lecturer>answering</lecturer>
        </Conditions>
    </Possibility>
    <Possibility>
        <Number>6</Number>
        <Type>Answer OK</Type>
        <NewState>3</NewState>
        <CutType>1</CutType>
        <Conditions>
            <lecturer>AnswerOK</lecturer>
        </Conditions>
    </Possibility>
</Possibilities>
</Event>
</Transitions>
</State>

<State>
    <Number>9</Number>
    <Name>Questioner PiP + Slide</Name>
    <Context>2</Context>
    <Camera>Slides</Camera>
    <CameraPiP>Audience</CameraPiP>
    <Transitions>
        <Time>
            <min>20</min>
            <recommend>30</recommend>
            <max>40</max>
            <randomRange>20%</randomRange>
        </Time>
        <Possibilities>
            <Possibility>
                <Number>1</Number>
                <Type>Time</Type>
                <NewState>13</NewState>
                <CutType>1,2</CutType>
                <Conditions>
                    <lecturer>active</lecturer>
                    <slide>active,annotate,switch</slide>
                    <questioner>inactive</questioner>
                </Conditions>
            </Possibility>
            <Possibility>
                <Number>2</Number>
                <Type>Time</Type>
                <NewState>7</NewState>
                <CutType>1,2</CutType>
                <Conditions>
                    <lecturer>inactive</lecturer>
                    <slide>inactive</slide>
                    <questioner>active</questioner>
                </Conditions>
            </Possibility>
            <Possibility>
                <Number>3</Number>
                <Type>Time</Type>
                <NewState>9</NewState>
                <CutType>5</CutType>
                <Conditions>
                    <questioner>active,still</questioner>
                </Conditions>
            </Possibility>
            <Possibility>
                <Number>4</Number>
                <Type>Time</Type>
                <NewState>9</NewState>
                <CutType>6</CutType>
                <Conditions>
                    <questioner>active,gesticulating</questioner>
                </Conditions>
            </Possibility>
        </Possibilities>
    </Time>
    <Event>
        <Possibilities>
            <Possibility>
                <Number>1</Number>
                <Type>Question Stopped</Type>
                <NewState>5</NewState>
                <CutType>1</CutType>
                <Conditions>
                    <questioner>denied,deferred</questioner>
                </Conditions>
            </Possibility>
            <Possibility>
                <Number>2</Number>
                <Type>End Of Lecture</Type>
                <NewState>15</NewState>
                <CutType>1</CutType>
                <Conditions>Empty</Conditions>
            </Possibility>
        </Possibilities>
    </Event>

```

```

    <Possibility>
      <Number>3</Number>
      <Type>Lecturer Answering</Type>
      <NewState>11</NewState>
      <CutType>1</CutType>
      <Conditions>
        <lecturer>answering</lecturer>
      </Conditions>
    </Possibility>
  <Possibility>
    <Number>4</Number>
    <Type>Lecturer Answering</Type>
    <NewState>12</NewState>
    <CutType>1</CutType>
    <Conditions>
      <lecturer>answering</lecturer>
    </Conditions>
  </Possibility>
  <Possibility>
    <Number>5</Number>
    <Type>Lecturer Answering</Type>
    <NewState>13</NewState>
    <CutType>1</CutType>
    <Conditions>
      <lecturer>answering,active,gesticulating,moving</lecturer>
      <slide>annotate,switch</slide>
    </Conditions>
  </Possibility>
  <Possibility>
    <Number>6</Number>
    <Type>Answer OK</Type>
    <NewState>5</NewState>
    <CutType>1</CutType>
    <Conditions>
      <lecturer>AnswerOK</lecturer>
    </Conditions>
  </Possibility>
</Possibilities>
</Event>
</Transitions>
</State>

<State>
  <Number>10</Number>
  <Name>Slide</Name>
  <Context>2</Context>
  <Camera>Slides</Camera>
  <CameraPiP>Empty</CameraPiP>
  <Transitions>
    <Time>
      <min>20</min>
      <recommend>30</recommend>
      <max>40</max>
      <randomRange>20%</randomRange>
    </Time>
    <Possibilities>
      <Possibility>
        <Number>1</Number>
        <Type>Time</Type>
        <NewState>14</NewState>
        <CutType>1,2</CutType>
        <Conditions>
          <lecturer>active</lecturer>
          <slide>active,annotate,switch</slide>
          <questioner>inactive</questioner>
        </Conditions>
      </Possibility>
      <Possibility>
        <Number>2</Number>
        <Type>Time</Type>
        <NewState>7</NewState>
        <CutType>1,2</CutType>
        <Conditions>
          <lecturer>inactive</lecturer>
          <slide>inactive</slide>
          <questioner>active</questioner>
        </Conditions>
      </Possibility>
    </Possibilities>
  </Time>
</Event>
  <Possibilities>
    <Possibility>
      <Number>1</Number>
      <Type>Question Stopped</Type>
      <NewState>6</NewState>
      <CutType>1</CutType>
      <Conditions>
        <questioner>denied,deferred</questioner>
      </Conditions>
    </Possibility>
    <Possibility>
      <Number>2</Number>
      <Type>End Of Lecture</Type>
      <NewState>15</NewState>
      <CutType>1</CutType>
      <Conditions>Empty</Conditions>
    </Possibility>
    <Possibility>
      <Number>3</Number>
      <Type>Lecturer Answering</Type>
      <NewState>11</NewState>

```

```

        <CutType>1</CutType>
        <Conditions>
          <lecturer>answering</lecturer>
        </Conditions>
      </Possibility>
    </Possibility>
    <Number>4</Number>
    <Type>Lecturer Answering</Type>
    <NewState>12</NewState>
    <CutType>1</CutType>
    <Conditions>
      <lecturer>answering</lecturer>
      <questioner>active</questioner>
    </Conditions>
  </Possibility>
</Possibility>
<Number>5</Number>
<Type>Lecturer Answering</Type>
<NewState>13</NewState>
<CutType>1</CutType>
<Conditions>
  <lecturer>answering,active,gesticulating,moving</lecturer>
  <slide>annotate,switch</slide>
</Conditions>
</Possibility>
</Possibility>
<Number>6</Number>
<Type>Lecturer Answering</Type>
<NewState>14</NewState>
<CutType>1</CutType>
<Conditions>
  <lecturer>answering,inactive,still</lecturer>
  <slide>annotate,switch</slide>
</Conditions>
</Possibility>
</Possibility>
<Number>7</Number>
<Type>Answer OK</Type>
<NewState>6</NewState>
<CutType>1</CutType>
<Conditions>
  <lecturer>AnswerOK</lecturer>
</Conditions>
</Possibility>
</Possibilities>
</Event>
</Transitions>
</State>

<State>
  <Number>11</Number>
  <Name>Lecturer</Name>
  <Context>3</Context>
  <Camera>Lecturer</Camera>
  <CameraPiP>Empty</CameraPiP>
  <Transitions>
    <Time>
      <min>15</min>
      <recommend>20</recommend>
      <max>30</max>
      <randomRange>20%</randomRange>
    </Time>
    <Possibilities>
      <Possibility>
        <Number>1</Number>
        <Type>Time</Type>
        <NewState>7</NewState>
        <CutType>1,2</CutType>
        <Conditions>
          <lecturer>inactive</lecturer>
          <questioner>active</questioner>
        </Conditions>
      </Possibility>
      <Possibility>
        <Number>2</Number>
        <Type>Time</Type>
        <NewState>14</NewState>
        <CutType>1,2</CutType>
        <Conditions>
          <lecturer>active</lecturer>
          <slide>active,noSpace,annotate,switch</slide>
          <questioner>inactive</questioner>
        </Conditions>
      </Possibility>
      <Possibility>
        <Number>3</Number>
        <Type>Time</Type>
        <NewState>13</NewState>
        <CutType>1,2</CutType>
        <Conditions>
          <lecturer>active</lecturer>
          <slide>active,space,annotate,switch</slide>
          <questioner>inactive</questioner>
        </Conditions>
      </Possibility>
      <Possibility>
        <Number>4</Number>
        <Type>Time</Type>
        <NewState>12</NewState>
        <CutType>1,2</CutType>
        <Conditions>
          <lecturer>active</lecturer>

```



```

        <slide>inactive</slide>
        <questioner>active</questioner>
    </Conditions>
</Possibility>
<Possibility>
    <Number>5</Number>
    <Type>Time</Type>
    <NewState>3</NewState>
    <CutType>1,2</CutType>
    <Conditions>
        <lecturer>active</lecturer>
        <questioner>inactive,time+10</questioner>
    </Conditions>
</Possibility>
<Possibility>
    <Number>6</Number>
    <Type>Time</Type>
    <NewState>11</NewState>
    <CutType>5</CutType>
    <Conditions>
        <lecturer>active,still</lecturer>
    </Conditions>
</Possibility>
<Possibility>
    <Number>7</Number>
    <Type>Time</Type>
    <NewState>11</NewState>
    <CutType>6</CutType>
    <Conditions>
        <lecturer>active,gesticulating</lecturer>
    </Conditions>
</Possibility>
</Possibilities>
</Time>
<Event>
    <Possibilities>
        <Possibility>
            <Number>1</Number>
            <Type>Answer OK</Type>
            <NewState>3</NewState>
            <CutType>1</CutType>
            <Conditions>
                <lecturer>AnswerOK</lecturer>
            </Conditions>
        </Possibility>
        <Possibility>
            <Number>2</Number>
            <Type>Answer Incomplete</Type>
            <NewState>7</NewState>
            <CutType>1</CutType>
            <Conditions>
                <lecturer>AnswerIncomplete</lecturer>
            </Conditions>
        </Possibility>
        <Possibility>
            <Number>3</Number>
            <Type>Lecturer Speaking</Type>
            <NewState>3</NewState>
            <CutType>1</CutType>
            <Conditions>
                <lecturer>speaking</lecturer>
            </Conditions>
        </Possibility>
        <Possibility>
            <Number>4</Number>
            <Type>End Of Lecture</Type>
            <NewState>15</NewState>
            <CutType>1</CutType>
            <Conditions>Empty</Conditions>
        </Possibility>
        <Possibility>
            <Number>5</Number>
            <Type>Question Acknowledged</Type>
            <NewState>7</NewState>
            <CutType>1</CutType>
            <Conditions>
                <questioner>acknowledged</questioner>
            </Conditions>
        </Possibility>
        <Possibility>
            <Number>6</Number>
            <Type>Slide Switch</Type>
            <NewState>14</NewState>
            <CutType>1</CutType>
            <Conditions>
                <slide>switch,annotate</slide>
            </Conditions>
        </Possibility>
        <Possibility>
            <Number>7</Number>
            <Type>Question Stopped</Type>
            <NewState>3</NewState>
            <CutType>1</CutType>
            <Conditions>
                <questioner>denied,deferred</questioner>
            </Conditions>
        </Possibility>
    <!-->
    <Possibility>
        <Number>8</Number>
        <Type>Slide Space</Type>
        <NewState>13</NewState>
    </Possibility>

```

```

        <CutType>1</CutType>
        <Conditions>
          <slide>space</slide>
        </Conditions>
      </Possibility>
    <Possibility>
      <Number>9</Number>
      <Type>Slide NoSpace</Type>
      <NewState>14</NewState>
      <CutType>1</CutType>
      <Conditions>
        <slide>nospace</slide>
      </Conditions>
    </Possibility>-->
  </Possibilities>
</Event>
</Transitions>
</State>

<State>
  <Number>12</Number>
  <Name>Lecturer PiP + Questioner</Name>
  <Context>3</Context>
  <Camera>Audience</Camera>
  <CameraPiP>Lecturer</CameraPiP>
  <Transitions>
    <Time>
      <min>15</min>
      <recommend>20</recommend>
      <max>30</max>
      <randomRange>20%</randomRange>
    <Possibilities>
      <Possibility>
        <Number>1</Number>
        <Type>Time</Type>
        <NewState>7</NewState>
        <CutType>1,2</CutType>
        <Conditions>
          <lecturer>inactive</lecturer>
          <slide>inactive</slide>
          <questioner>active</questioner>
        </Conditions>
      </Possibility>
      <Possibility>
        <Number>2</Number>
        <Type>Time</Type>
        <NewState>13</NewState>
        <CutType>1,2</CutType>
        <Conditions>
          <lecturer>active</lecturer>
          <slide>active,space,annotate,switch</slide>
          <questioner>inactive</questioner>
        </Conditions>
      </Possibility>
      <Possibility>
        <Number>3</Number>
        <Type>Time</Type>
        <NewState>14</NewState>
        <CutType>1,2</CutType>
        <Conditions>
          <lecturer>active</lecturer>
          <slide>active,nospace,annotate,switch</slide>
          <questioner>inactive</questioner>
        </Conditions>
      </Possibility>
      <Possibility>
        <Number>4</Number>
        <Type>Time</Type>
        <NewState>3</NewState>
        <CutType>1,2</CutType>
        <Conditions>
          <lecturer>active</lecturer>
          <questioner>inactive,time+05</questioner>
        </Conditions>
      </Possibility>
      <Possibility>
        <Number>5</Number>
        <Type>Time</Type>
        <NewState>11</NewState>
        <CutType>1,2</CutType>
        <Conditions>
          <lecturer>active</lecturer>
          <slide>inactive</slide>
          <questioner>inactive</questioner>
        </Conditions>
      </Possibility>
      <Possibility>
        <Number>6</Number>
        <Type>Time</Type>
        <NewState>12</NewState>
        <CutType>5</CutType>
        <Conditions>
          <lecturer>active,still</lecturer>
          <questioner>active,still</questioner>
        </Conditions>
      </Possibility>
      <Possibility>
        <Number>7</Number>
        <Type>Time</Type>
        <NewState>12</NewState>
        <CutType>6</CutType>

```

```

        <Conditions>
        <lecturer>active,gesticulating</lecturer>
        <questioner>active,gesticulating</questioner>
        </Conditions>
    </Possibility>
</Possibilities>
</Time>
<Event>
    <Possibilities>
        <Possibility>
            <Number>1</Number>
            <Type>Answer OK</Type>
            <NewState>3</NewState>
            <CutType>1</CutType>
            <Conditions>
                <lecturer>AnswerOK</lecturer>
            </Conditions>
        </Possibility>
        <Possibility>
            <Number>2</Number>
            <Type>Answer Incomplete</Type>
            <NewState>7</NewState>
            <CutType>1</CutType>
            <Conditions>
                <lecturer>AnswerIncomplete</lecturer>
            </Conditions>
        </Possibility>
        <Possibility>
            <Number>3</Number>
            <Type>Lecturer Speaking</Type>
            <NewState>3</NewState>
            <CutType>1</CutType>
            <Conditions>
                <lecturer>speaking</lecturer>
            </Conditions>
        </Possibility>
        <Possibility>
            <Number>4</Number>
            <Type>End Of Lecture</Type>
            <NewState>15</NewState>
            <CutType>1</CutType>
            <Conditions>Empty</Conditions>
        </Possibility>
        <Possibility>
            <Number>5</Number>
            <Type>Question Acknowledged</Type>
            <NewState>7</NewState>
            <CutType>1</CutType>
            <Conditions>
                <questioner>acknowledged</questioner>
            </Conditions>
        </Possibility>
        <Possibility>
            <Number>6</Number>
            <Type>Slide Switch</Type>
            <NewState>14</NewState>
            <CutType>1</CutType>
            <Conditions>
                <slide>switch,annotate</slide>
            </Conditions>
        </Possibility>
        <Possibility>
            <Number>7</Number>
            <Type>Question Stopped</Type>
            <NewState>3</NewState>
            <CutType>1</CutType>
            <Conditions>
                <questioner>denied,deferred</questioner>
            </Conditions>
        </Possibility>
        <!--Possibility>
            <Number>8</Number>
            <Type>Slide Space</Type>
            <NewState>13</NewState>
            <CutType>1</CutType>
            <Conditions>
                <slide>space</slide>
            </Conditions>
        </Possibility>
        <Possibility>
            <Number>9</Number>
            <Type>Slide NoSpace</Type>
            <NewState>14</NewState>
            <CutType>1</CutType>
            <Conditions>
                <slide>nospace</slide>
            </Conditions>
        </Possibility>-->
    </Possibilities>
</Event>
</Transitions>
</State>

<State>
    <Number>13</Number>
    <Name>Lecturer PiP + Slide</Name>
    <Context>3</Context>
    <Camera>Slides</Camera>
    <CameraPiP>Lecturer</CameraPiP>
    <Transitions>
    <Time>

```

```

<min>20</min>
<recommend>25</recommend>
<max>30</max>
<randomRange>20%</randomRange>
<Possibilities>
  <Possibility>
    <Number>1</Number>
    <Type>Time</Type>
    <NewState>5</NewState>
    <CutType>1,2</CutType>
    <Conditions>
      <lecturer>active</lecturer>
      <slide>active,space,annotate,switch</slide>
      <questioner>inactive,time+10</questioner>
    </Conditions>
  </Possibility>
  <Possibility>
    <Number>2</Number>
    <Type>Time</Type>
    <NewState>14</NewState>
    <CutType>1,2</CutType>
    <Conditions>
      <lecturer>active</lecturer>
      <slide>active,noSpace,annotate,switch</slide>
    </Conditions>
  </Possibility>
  <Possibility>
    <Number>3</Number>
    <Type>Time</Type>
    <NewState>12</NewState>
    <CutType>1,2</CutType>
    <Conditions>
      <lecturer>active</lecturer>
      <questioner>active</questioner>
    </Conditions>
  </Possibility>
  <Possibility>
    <Number>4</Number>
    <Type>Time</Type>
    <NewState>7</NewState>
    <CutType>1,2</CutType>
    <Conditions>
      <lecturer>inactive</lecturer>
      <questioner>active</questioner>
    </Conditions>
  </Possibility>
  <Possibility>
    <Number>5</Number>
    <Type>Time</Type>
    <NewState>11</NewState>
    <CutType>1,2</CutType>
    <Conditions>
      <lecturer>active</lecturer>
      <slide>inactive</slide>
      <questioner>inactive</questioner>
    </Conditions>
  </Possibility>
  <Possibility>
    <Number>6</Number>
    <Type>Time</Type>
    <NewState>13</NewState>
    <CutType>5</CutType>
    <Conditions>
      <lecturer>active,still</lecturer>
    </Conditions>
  </Possibility>
  <Possibility>
    <Number>7</Number>
    <Type>Time</Type>
    <NewState>13</NewState>
    <CutType>6</CutType>
    <Conditions>
      <lecturer>active,gesticulating</lecturer>
    </Conditions>
  </Possibility>
</Possibilities>
</Time>
<Event>
  <Possibilities>
    <Possibility>
      <Number>1</Number>
      <Type>Answer OK</Type>
      <NewState>5</NewState>
      <CutType>1</CutType>
      <Conditions>
        <lecturer>AnswerOK</lecturer>
      </Conditions>
    </Possibility>
    <Possibility>
      <Number>2</Number>
      <Type>Answer Incomplete</Type>
      <NewState>7</NewState>
      <CutType>1</CutType>
      <Conditions>
        <lecturer>AnswerIncomplete</lecturer>
      </Conditions>
    </Possibility>
    <Possibility>
      <Number>3</Number>
      <Type>Lecturer Speaking</Type>
      <NewState>5</NewState>

```

```

        <CutType>1</CutType>
        <Conditions>
            <lecturer>speaking</lecturer>
        </Conditions>
    </Possibility>
</Possibility>
<Possibility>
    <Number>4</Number>
    <Type>End Of Lecture</Type>
    <NewState>15</NewState>
    <CutType>1</CutType>
    <Conditions>Empty</Conditions>
</Possibility>
<Possibility>
    <Number>5</Number>
    <Type>Question Acknowledged</Type>
    <NewState>7</NewState>
    <CutType>1</CutType>
    <Conditions>
        <questioner>acknowledged</questioner>
    </Conditions>
</Possibility>
<Possibility>
    <Number>6</Number>
    <Type>Question Stopped</Type>
    <NewState>5</NewState>
    <CutType>1</CutType>
    <Conditions>
        <questioner>denied,deferred</questioner>
    </Conditions>
</Possibility>
<!--<Possibility>
    <Number>7</Number>
    <Type>Slide NoSpace</Type>
    <NewState>14</NewState>
    <CutType>1</CutType>
    <Conditions>
        <slide>nospace</slide>
    </Conditions>
</Possibility>-->
</Possibilities>
</Event>
</Transitions>
</State>

<State>
    <Number>14</Number>
    <Name>Slide</Name>
    <Context>3</Context>
    <Camera>Slides</Camera>
    <CameraPiP>Empty</CameraPiP>
    <Transitions>
        <Time>
            <min>20</min>
            <recommend>25</recommend>
            <max>30</max>
            <randomRange>20%</randomRange>
        </Time>
        <Possibilities>
            <Possibility>
                <Number>1</Number>
                <Type>Time</Type>
                <NewState>6</NewState>
                <CutType>1,2</CutType>
                <Conditions>
                    <lecturer>active</lecturer>
                    <slide>active,nospace,annotate,switch</slide>
                    <questioner>inactive,time+05</questioner>
                </Conditions>
            </Possibility>
            <Possibility>
                <Number>2</Number>
                <Type>Time</Type>
                <NewState>13</NewState>
                <CutType>1,2</CutType>
                <Conditions>
                    <lecturer>active</lecturer>
                    <slide>active,nospace,annotate,switch</slide>
                </Conditions>
            </Possibility>
            <Possibility>
                <Number>3</Number>
                <Type>Time</Type>
                <NewState>11</NewState>
                <CutType>1,2</CutType>
                <Conditions>
                    <lecturer>active</lecturer>
                    <slide>inactive</slide>
                </Conditions>
            </Possibility>
            <Possibility>
                <Number>4</Number>
                <Type>Time</Type>
                <NewState>7</NewState>
                <CutType>1,2</CutType>
                <Conditions>
                    <lecturer>inactive</lecturer>
                    <questioner>active</questioner>
                </Conditions>
            </Possibility>
        </Possibilities>
    </Time>
</Event>

```

```

<Possibilities>
  <Possibility>
    <Number>1</Number>
    <Type>Answer OK</Type>
    <NewState>6</NewState>
    <CutType>1</CutType>
    <Conditions>
      <lecturer>AnswerOK</lecturer>
    </Conditions>
  </Possibility>
  <Possibility>
    <Number>2</Number>
    <Type>Answer Incomplete</Type>
    <NewState>7</NewState>
    <CutType>1</CutType>
    <Conditions>
      <lecturer>AnswerIncomplete</lecturer>
    </Conditions>
  </Possibility>
  <Possibility>
    <Number>3</Number>
    <Type>Lecturer Speaking</Type>
    <NewState>6</NewState>
    <CutType>1</CutType>
    <Conditions>
      <lecturer>speaking</lecturer>
    </Conditions>
  </Possibility>
  <Possibility>
    <Number>4</Number>
    <Type>End Of Lecture</Type>
    <NewState>15</NewState>
    <CutType>1</CutType>
    <Conditions>Empty</Conditions>
  </Possibility>
  <Possibility>
    <Number>5</Number>
    <Type>Question Acknowledged</Type>
    <NewState>7</NewState>
    <CutType>1</CutType>
    <Conditions>
      <questioner>acknowledged</questioner>
    </Conditions>
  </Possibility>
  <Possibility>
    <Number>6</Number>
    <Type>Question Stopped</Type>
    <NewState>6</NewState>
    <CutType>1</CutType>
    <Conditions>
      <questioner>denied,deferred</questioner>
    </Conditions>
  </Possibility>
  <!--Possibility>
    <Number>7</Number>
    <Type>Slide Space</Type>
    <NewState>13</NewState>
    <CutType>1</CutType>
    <Conditions>
      <slide>space</slide>
    </Conditions>
  </Possibility-->
</Possibilities>
</Event>
</Transitions>
</State>

<State>
  <Number>15</Number>
  <Name>End</Name>
  <Context>0</Context>
  <Camera>LongShot</Camera>
  <CameraPiP>Empty</CameraPiP>
  <Transitions>
  </Transitions>
</State>

</Definition>

</FSM>

```

7.1.2. Example Configuration File of the Cameraman

```

<?xml version="1.0" encoding="utf-8" ?>
<cameraman>

  <generalinformation>
    <cameramannname>Axis1_Lecturer</cameramannname>
    <cameramanipaddress>134.155.92.37</cameramanipaddress>
    <cameratarget>lecturer</cameratarget>
    <samplesize>352x288</samplesize>
    <numberofsamplespersec>10</numberofsamplespersec>
    <delayforcontrolloop>550</delayforcontrolloop>
  </generalinformation>

  <director>
    <address>134.155.92.68</address>
    <port>55555</port>
  </director>

```

```

    <numberofreconnects>10</numberofreconnects>
    <messagequeuesize>15</messagequeuesize>
</director>

<camera>
  <general>
    <cameratyp>axis</cameratyp>
    <control>ptz</control>
    <streamingaddress>255.255.255.255</streamingaddress>
  </general>

  <webcam>
    <address>134.155.92.23</address>
    <username></username>
    <password></password>
  </webcam>

  <localcam>
    <comport></comport>
  </localcam>

  <ptz>
    <cameraposition>
      <x>6,66</x>
      <y>2,75</y>
      <z>1,45</z>
      <side>False</side>
    </cameraposition>

    <homeposition>
      <pan>-26</pan>
      <tilt>-7</tilt>
      <zoom>5000</zoom>
    </homeposition>

    <moveablerange>
      <panleft>100</panleft>
      <panright>100</panright>
      <tiltup>90</tiltup>
      <tiltdown>30</tiltdown>
    </moveablerange>

  </ptz>

  <technicaldata>
    <ccdwidth>0,0036</ccdwidth>
    <focalwidth>0,0041</focalwidth>
    <maxzoomfactor>18</maxzoomfactor>
    <zoomvalues>
      <min>1</min>
      <max>9999</max>
    </zoomvalues>

  </technicaldata>
</camera>

<control>
  <shot>
    <arrangementpoint>
      <x>92</x>
      <y>130</y>
    </arrangementpoint>

    <targetwidth>0,8</targetwidth>
    <zoomoutfactor>1,7</zoomoutfactor>
    <persondetection>
      <numberofslices>16</numberofslices>
      <numberofsamples>3</numberofsamples>
      <matchingsamples>2</matchingsamples>
      <motiondetectionthreshold>10</motiondetectionthreshold>
      <skindetectionthreshold>30</skindetectionthreshold>
      <bandpassfilterwidth>2</bandpassfilterwidth>
      <areasizethreshold>3</areasizethreshold>
      <facedetectiongradient>0,9</facedetectiongradient>
    </persondetection>

    <motiondetection>
      <numberofsamples>3</numberofsamples>
      <numberofslices>22</numberofslices>
      <motiondetectionsensitivity>30</motiondetectionsensitivity>
      <bandpassfilterwidth>5</bandpassfilterwidth>
      <motionlevelthreshold>15</motionlevelthreshold>
    </motiondetection>

  </shot>

  <iriscontrol>
    <minimumbrightness>80</minimumbrightness>
    <maximumbrightness>220</maximumbrightness>
    <minimumbrightnesscheck>90</minimumbrightnesscheck>
    <maximumbrightnesscheck>150</maximumbrightnesscheck>
    <usepersondetection>false</usepersondetection>
    <motiondetection>
      <numberofslices>22</numberofslices>
      <numberofsamples>3</numberofsamples>
      <motiondetectionsensitivity>50</motiondetectionsensitivity>
      <bandpassfilterwidth>2</bandpassfilterwidth>
      <matchingsamples>2</matchingsamples>
    </motiondetection>
  </iriscontrol>
</control>

```

```

        </iriscontrol>

    </control>

</cameraman>

```

7.2. Sourcecode Snippets

7.2.1. Function “FSM.GetTimerInterval”

```

Private Function GetTimerInterval(ByVal minSec As Integer, _
                                ByVal recommendSec As Integer, _
                                ByVal minRange As Integer, _
                                ByVal maxRange As Integer, _
                                ByVal addTime As String) As Double

    Dim result, myMin, myMax As Double
    Dim Seconds As Integer

    myMax = (maxRange - minRange) * Rnd() + minRange
    myMin = (recommendSec - minSec) * Rnd() + minSec
    result = CDBl(Int((myMax - myMin) * Rnd() + myMin))

    Seconds = CInt(Right(addTime, 2))
    Select Case Left(addTime, 1)
        Case "+"
            result = result + Seconds
        Case "-"
            result = result - Seconds
        Case "="
            If Seconds > 0 Then
                result = Seconds
            End If
    End Select
    result = Math.Max(result, 4)

    'return result in milliseconds
    Return (result * 1000)
End Function

```

7.2.2. Procedure “AVMixer.startgettingFrames”

```

public void startgettingFrames()
{
    int got_picture = 0;

    // Allocate video frame
    vFrame = FFmpeg.avcodec_alloc_frame();

    // Allocate memory for packet
    IntPtr pPacket = Allocate<FFmpeg.AVPacket>();

    while (FFmpeg.av_read_frame(pFormatContext, pPacket) >= 0)
    {
        // Get packet from pointer
        FFmpeg.AVPacket packet = PtrToStructure<FFmpeg.AVPacket>(pPacket);

        #region video stream
        // Is this a packet from the video stream?
        if (packet.stream_index == videoStream.index)
        {
            // Decode video frame
            int length = FFmpeg.avcodec_decode_video(videoStream.codec, vFrame, ref got_picture, packet.data, packet.size);

            // Did we get a video frame?
            if (got_picture != 0)
            {
                LatestFrame = YUV2RGB(PtrToStructure<FFmpeg.AVFrame>(vFrame), 720, 576);

                //event senden
                this.myFrameReady.Invoke();
            }
            break;
        }
        #endregion

        #region audio stream
        if (packet.stream_index == audioStream.index)
        {
            aFrame = new byte[aFrameSize];
            paFrame = Marshal.UnsafeAddrOfPinnedArrayElement(aFrame, 0);

            //Decode audio frame
            int length = FFmpeg.avcodec_decode_audio(audioStream.codec, paFrame, ref aFrameSize, packet.data, packet.size);

            //did we get an audio frame?
            if (length > 0)
            {
                this.AudioReceived(aFrame);
            }
        }

        aFrame = null;
    }
}

```



```

        break;
    }
    #endregion
    System.Windows.Forms.Application.DoEvents();
}

// Free the packet that was allocated by av_read_frame
FFmpeg.av_free_packet(pPacket);

// Free memory
FFmpeg.av_free(vFrame);
}

```

7.2.3. Function “DefineCompressionLine”

```

public double[,] DefineCompressionLine(double StartIn, double StartOut, double EndNoiseGateIn,
    double EndNoiseGateOut, double EndCompressionIn, double EndCompressionOut,
    double EndLimitingIn, double EndLimitingOut)
{
    double[,] result = new double[2,4];
    result[0, 0] = StartIn;
    result[1, 0] = StartOut;
    result[0, 1] = EndNoiseGateIn;
    result[1, 1] = EndNoiseGateOut;
    result[0, 2] = EndCompressionIn;
    result[1, 2] = EndCompressionOut;
    result[0, 3] = EndLimitingIn;
    result[1, 3] = EndLimitingOut;
    return result;
}

```

7.2.4. Function “Sample2DB”

```

public double Sample2DB(Int16 AbsoluteSampleValue)
{
    if (AbsoluteSampleValue == 0)
        AbsoluteSampleValue = 1;
    double result = 20 * Math.Log10(AbsoluteSampleValue / Int16.MaxValue);
    return result;
}

```

7.2.5. Function “DB2SampleValue”

```

public Int16 DB2SampleValue(double dB)
{
    Int16 result = (Int16)Math.Round(Int16.MaxValue * Math.Pow(10d, (dB / 20)));
    return result;
}

```

7.2.6. Function “WaveExtrema”

```

private Int16[] WaveExtrema(Int16[] myPCM, Int16 carryOver)
{
    Int16 Extremum = Math.Abs(carryOver);
    Int16 v1, v2;
    bool DirectionUP;
    Int16[] result = new Int16[myPCM.Length];

    if (Extremum < Math.Abs(myPCM[0]))
        DirectionUP = true;
    else
        DirectionUP = false;

    for (int count = -1; count < myPCM.Length - 1; count++)
    {
        if (count == -1)
            v1 = Math.Abs(carryOver);
        else

```

```

        v1 = Math.Abs(myPCM[count]);
        v2 = Math.Abs(myPCM[count + 1]);

        if (DirectionUP)
        {
            if (v1 >= v2)
            {
                Extremum = v1;
                DirectionUP = false;
            }
        }
        else
        {
            if (v1 < v2)
            {
                DirectionUP = true;
            }
        }
        if (count > -1)
        {
            result[count] = Extremum;
        }
    }

    return result;
}

```

7.2.7. Function “getFactor”

```

private double getFactor(Int16 ExtremaIn, double[,] usedCompLine)
{
    double dBIn = Sample2DB(ExtremaIn);
    double dbOut = 0;
    Int16 SampleOut = 0;
    double result = 0;
    double x1, x2, y1, y2;
    double m, b;
    if (dBIn < usedCompLine[0, 1]) //NoiseGatePhase
    {
        x1 = usedCompLine[0, 0];
        x2 = usedCompLine[0, 1];
        y1 = usedCompLine[1, 0];
        y2 = usedCompLine[1, 1];
    }
    else if (dBIn < usedCompLine[0, 2]) //CompressionPhase
    {
        x1 = usedCompLine[0, 1];
        x2 = usedCompLine[0, 2];
        y1 = usedCompLine[1, 1];
        y2 = usedCompLine[1, 2];
    }
    else //LimitingPhase
    {
        x1 = usedCompLine[0, 2];
        x2 = usedCompLine[0, 3];
        y1 = usedCompLine[1, 2];
        y2 = usedCompLine[1, 3];
    }
    //linear interpolation function parameters
    m = ((double)(y2 - y1) / (double)(x2 - x1));
    b = y2 - (m * x2);
    //linear interpolation: f(x) = mx + b
    dbOut = (double)(m * dBIn) + b;
    SampleOut = DB2SampleValue(dbOut);
    result = SampleOut / ExtremaIn;
    return result;
}

```

7.3. Original Evaluation Papers

7.3.1. German Pre-Test Form

<p>FAKULTÄT FÜR SOZIALWISSENSCHAFTEN Lehrstuhl Erziehungswissenschaft I Prof. Dr. Peter Drewek</p> <p>INSTITUT FÜR INFORMATIK Lehrstuhl für Praktische Informatik IV Prof. Dr. Wolfgang Effelsberg</p> <p>UNIVERSITÄT MANNHEIM</p> <table border="1" style="margin: 10px auto; width: 150px;"> <tr> <td style="width: 50%;">Video-Nr.</td> <td style="width: 50%;">VPN - Nr.</td> </tr> </table> <p>Vorfragen zum Video</p> <p>Die Beantwortung der nachstehenden Fragen zur Ermittlung Ihrer Kenntnisse im Themengebiet „Audioaufnahme und Audioschnitte für Videoproduktionen“ erfolgen anonym und dienen uns lediglich zur Erfassung Ihres Wissens in diesem Bereich.</p> <ol style="list-style-type: none"> 1. Was ist der Unterschied zwischen der Samplingtiefe und der Samplerate? 2. In welcher Einheit wird die Samplingtiefe angegeben? 3. Nennen Sie die typischen Eigenschaften von qualitativ hochwertigem DV-Video. 4. Nennen Sie zwei verschiedene Mikrofon-Charakteristiken. 5. Was benötigen Elektret-Kondensatormikrofone? 6. Was ist der Unterschied zwischen einem Popp-Schutz und einem Windschutz? 7. Wie viele interviewte Personen sollen maximal mit einem Mikrofon in einem Interview aufgenommen werden? 	Video-Nr.	VPN - Nr.	<ol style="list-style-type: none"> 8. Was ist der Unterschied zwischen einer Mikrofonangel und einem Mikrofonstativ? 9. Was ist eine Atmo-Spur? 10. Wozu wird eine Atmo-Spur benötigt? 11. Wie heisst der besprochene Effekt, der eintreten kann, wenn ein Signal mit mehreren Mikrofonen aufgenommen wird? 12. Wie laut soll der lauteste Ton aufgenommen werden? (in dB) 13. Wie laut soll der leiseste, gut hörbare, Ton aufgenommen werden? (in dB) 14. Nennen Sie ein Beispiel, wann eine Ton-Überblendung bei einem Szenenwechsel sinnvoller ist, als ein harter Schnitt?
Video-Nr.	VPN - Nr.		

7.3.2. German Questionnaire

<p>FAKULTÄT FÜR SOZIALWISSENSCHAFTEN Lehrstuhl Erziehungswissenschaft I Prof. Dr. Peter Drewek</p> <p>INSTITUT FÜR INFORMATIK Lehrstuhl für Praktische Informatik IV Prof. Dr. Wolfgang Effelsberg</p> <p>UNIVERSITÄT MANNHEIM</p> <table border="1" style="margin: 10px auto; width: 150px;"> <tr> <td style="width: 50%;">Video -Nr.</td> <td style="width: 50%;">VPN - Nr.</td> </tr> </table> <p>Fragebogen zum Video</p> <p>Uns interessiert Ihre Meinung zu dem Vorlesungsvideo, welches Sie sich gerade angeschaut haben. Hierzu möchten wir Ihnen nachstehend einige Fragen vorlegen. Diese Fragen stellen keinen Test dar, aus diesem Grund gibt es auch keine richtigen oder falschen Antworten. Diese Umfrage wird anonym durchgeführt. Aus diesem Grund möchten wir Sie bitten, die folgenden Fragen nach besten Wissen und Gewissen zu beantworten.</p> <p>Beachten Sie bitte beim Ausfüllen des Fragebogens darauf:</p> <ul style="list-style-type: none"> - Bei den meisten Fragen sind mehrere Antwortmöglichkeiten genannt, von denen Sie immer nur die eine Antwort, die für Sie zutrifft, ankreuzen sollen. Bitte kreuzen Sie in jeder Zeile nur eine Antwort an. - Es ist sehr wichtig für uns, dass wir Ihre persönlichen Ansichten erfahren; deshalb bitten wir Sie, den Fragebogen alleine auszufüllen. <p><u>Demografische Daten:</u></p> <p>(01) Wie alt sind Sie? Ich bin _____ Jahre.</p> <p>(02) Welches Geschlecht haben Sie? männlich <input type="radio"/> (1) weiblich <input type="radio"/> (2)</p> <p>(03) Welchen Studiengang studieren Sie: _____</p> <p>(04) In welchem Semester befinden Sie sich? Ich bin im _____ Semester.</p>	Video -Nr.	VPN - Nr.	<p>Wir würden gerne etwas zu Ihrer momentanen Einstellung hinsichtlich des angeschauten Videos erfahren. Dazu finden Sie auf dieser Seite einige Aussagen. Bitte kreuzen Sie die Zahl an, welche am Besten auf Sie zutrifft. Beachten Sie bitte, dass Sie in jeder Zeile nur einer Aussage zustimmen dürfen.</p> <table style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 5%;"></th> <th style="width: 15%; text-align: center;">Trifft nicht zu</th> <th style="width: 70%; text-align: center;">.....</th> <th style="width: 10%; text-align: center;">Trifft voll zu</th> </tr> </thead> <tbody> <tr> <td>(5) a) Ich mag diese Art des Videos. (I)</td> <td style="text-align: center;"><input type="radio"/> (1)</td> <td style="text-align: center;"><input type="radio"/> (2) <input type="radio"/> (3) <input type="radio"/> (4) <input type="radio"/> (5) <input type="radio"/> (6) <input type="radio"/> (7)</td> <td style="text-align: center;"><input type="radio"/> (7)</td> </tr> <tr> <td>b) Ich glaube, dass ich mit Hilfe des Videos Fragen zur Vorlesungssitzung gewachsen sein werde. (E)</td> <td style="text-align: center;"><input type="radio"/> (1)</td> <td style="text-align: center;"><input type="radio"/> (2) <input type="radio"/> (3) <input type="radio"/> (4) <input type="radio"/> (5) <input type="radio"/> (6) <input type="radio"/> (7)</td> <td style="text-align: center;"><input type="radio"/> (7)</td> </tr> <tr> <td>c) Wahrscheinlich werde ich inhaltliche Fragen zur Vorlesungssitzung, die auf dem Video basieren, nicht schaffen. (E)</td> <td style="text-align: center;"><input type="radio"/> (1)</td> <td style="text-align: center;"><input type="radio"/> (2) <input type="radio"/> (3) <input type="radio"/> (4) <input type="radio"/> (5) <input type="radio"/> (6) <input type="radio"/> (7)</td> <td style="text-align: center;"><input type="radio"/> (7)</td> </tr> <tr> <td>d) Ich fühle mich unter Druck bei diesem Video gut aufpassen zu müssen. (M)</td> <td style="text-align: center;"><input type="radio"/> (1)</td> <td style="text-align: center;"><input type="radio"/> (2) <input type="radio"/> (3) <input type="radio"/> (4) <input type="radio"/> (5) <input type="radio"/> (6) <input type="radio"/> (7)</td> <td style="text-align: center;"><input type="radio"/> (7)</td> </tr> <tr> <td>e) Dieses Video war eine richtige Herausforderung für mich. (H)</td> <td style="text-align: center;"><input type="radio"/> (1)</td> <td style="text-align: center;"><input type="radio"/> (2) <input type="radio"/> (3) <input type="radio"/> (4) <input type="radio"/> (5) <input type="radio"/> (6) <input type="radio"/> (7)</td> <td style="text-align: center;"><input type="radio"/> (7)</td> </tr> <tr> <td>f) Nach dem Anschauen des Videos erscheint mir der Lernstoff sehr interessant. (I)</td> <td style="text-align: center;"><input type="radio"/> (1)</td> <td style="text-align: center;"><input type="radio"/> (2) <input type="radio"/> (3) <input type="radio"/> (4) <input type="radio"/> (5) <input type="radio"/> (6) <input type="radio"/> (7)</td> <td style="text-align: center;"><input type="radio"/> (7)</td> </tr> <tr> <td>g) Ich bin sehr gespannt darauf, wie gut ich bei diesem Video aufgepasst habe. (H)</td> <td style="text-align: center;"><input type="radio"/> (1)</td> <td style="text-align: center;"><input type="radio"/> (2) <input type="radio"/> (3) <input type="radio"/> (4) <input type="radio"/> (5) <input type="radio"/> (6) <input type="radio"/> (7)</td> <td style="text-align: center;"><input type="radio"/> (7)</td> </tr> <tr> <td>h) Ich fürchte mich ein wenig davor, dass ich mich bei inhaltlichen Fragen zu diesem Video blamieren könnte. (M)</td> <td style="text-align: center;"><input type="radio"/> (1)</td> <td style="text-align: center;"><input type="radio"/> (2) <input type="radio"/> (3) <input type="radio"/> (4) <input type="radio"/> (5) <input type="radio"/> (6) <input type="radio"/> (7)</td> <td style="text-align: center;"><input type="radio"/> (7)</td> </tr> <tr> <td>i) Ich war fest entschlossen, mich bei diesem Video voll anzustrengen. (H)</td> <td style="text-align: center;"><input type="radio"/> (1)</td> <td style="text-align: center;"><input type="radio"/> (2) <input type="radio"/> (3) <input type="radio"/> (4) <input type="radio"/> (5) <input type="radio"/> (6) <input type="radio"/> (7)</td> <td style="text-align: center;"><input type="radio"/> (7)</td> </tr> <tr> <td>j) Bei einem Video wie diesem brauche ich keine Belohnung, es macht mir auch so Spaß. (I)</td> <td style="text-align: center;"><input type="radio"/> (1)</td> <td style="text-align: center;"><input type="radio"/> (2) <input type="radio"/> (3) <input type="radio"/> (4) <input type="radio"/> (5) <input type="radio"/> (6) <input type="radio"/> (7)</td> <td style="text-align: center;"><input type="radio"/> (7)</td> </tr> <tr> <td>k) Es ist mir etwas peinlich, bei inhaltlichen Fragen zu diesem Video zu versagen. (M)</td> <td style="text-align: center;"><input type="radio"/> (1)</td> <td style="text-align: center;"><input type="radio"/> (2) <input type="radio"/> (3) <input type="radio"/> (4) <input type="radio"/> (5) <input type="radio"/> (6) <input type="radio"/> (7)</td> <td style="text-align: center;"><input type="radio"/> (7)</td> </tr> <tr> <td>l) Ich glaube mit Hilfe dieses Videos kann jeder lernen. (E)</td> <td style="text-align: center;"><input type="radio"/> (1)</td> <td style="text-align: center;"><input type="radio"/> (2) <input type="radio"/> (3) <input type="radio"/> (4) <input type="radio"/> (5) <input type="radio"/> (6) <input type="radio"/> (7)</td> <td style="text-align: center;"><input type="radio"/> (7)</td> </tr> <tr> <td>m) Ich glaube, ich kann mit diesem Video nicht lernen. (E)</td> <td style="text-align: center;"><input type="radio"/> (1)</td> <td style="text-align: center;"><input type="radio"/> (2) <input type="radio"/> (3) <input type="radio"/> (4) <input type="radio"/> (5) <input type="radio"/> (6) <input type="radio"/> (7)</td> <td style="text-align: center;"><input type="radio"/> (7)</td> </tr> <tr> <td>n) Wenn ich mit Hilfe dieses Videos etwas lerne, werde ich schon ein wenig stolz auf meine Tüchtigkeit sein. (H)</td> <td style="text-align: center;"><input type="radio"/> (1)</td> <td style="text-align: center;"><input type="radio"/> (2) <input type="radio"/> (3) <input type="radio"/> (4) <input type="radio"/> (5) <input type="radio"/> (6) <input type="radio"/> (7)</td> <td style="text-align: center;"><input type="radio"/> (7)</td> </tr> <tr> <td>o) Wenn ich an dieses Video denke, bin ich etwas beunruhigt. (M)</td> <td style="text-align: center;"><input type="radio"/> (1)</td> <td style="text-align: center;"><input type="radio"/> (2) <input type="radio"/> (3) <input type="radio"/> (4) <input type="radio"/> (5) <input type="radio"/> (6) <input type="radio"/> (7)</td> <td style="text-align: center;"><input type="radio"/> (7)</td> </tr> <tr> <td>p) Mit dieser Art von Video würde ich mich auch in meiner Freizeit beschäftigen. (I)</td> <td style="text-align: center;"><input type="radio"/> (1)</td> <td style="text-align: center;"><input type="radio"/> (2) <input type="radio"/> (3) <input type="radio"/> (4) <input type="radio"/> (5) <input type="radio"/> (6) <input type="radio"/> (7)</td> <td style="text-align: center;"><input type="radio"/> (7)</td> </tr> <tr> <td>q) Die Leistungsanforderungen aus diesem Video lähmen mich. (M)</td> <td style="text-align: center;"><input type="radio"/> (1)</td> <td style="text-align: center;"><input type="radio"/> (2) <input type="radio"/> (3) <input type="radio"/> (4) <input type="radio"/> (5) <input type="radio"/> (6) <input type="radio"/> (7)</td> <td style="text-align: center;"><input type="radio"/> (7)</td> </tr> </tbody> </table>		Trifft nicht zu	Trifft voll zu	(5) a) Ich mag diese Art des Videos. (I)	<input type="radio"/> (1)	<input type="radio"/> (2) <input type="radio"/> (3) <input type="radio"/> (4) <input type="radio"/> (5) <input type="radio"/> (6) <input type="radio"/> (7)	<input type="radio"/> (7)	b) Ich glaube, dass ich mit Hilfe des Videos Fragen zur Vorlesungssitzung gewachsen sein werde. (E)	<input type="radio"/> (1)	<input type="radio"/> (2) <input type="radio"/> (3) <input type="radio"/> (4) <input type="radio"/> (5) <input type="radio"/> (6) <input type="radio"/> (7)	<input type="radio"/> (7)	c) Wahrscheinlich werde ich inhaltliche Fragen zur Vorlesungssitzung, die auf dem Video basieren, nicht schaffen. (E)	<input type="radio"/> (1)	<input type="radio"/> (2) <input type="radio"/> (3) <input type="radio"/> (4) <input type="radio"/> (5) <input type="radio"/> (6) <input type="radio"/> (7)	<input type="radio"/> (7)	d) Ich fühle mich unter Druck bei diesem Video gut aufpassen zu müssen. (M)	<input type="radio"/> (1)	<input type="radio"/> (2) <input type="radio"/> (3) <input type="radio"/> (4) <input type="radio"/> (5) <input type="radio"/> (6) <input type="radio"/> (7)	<input type="radio"/> (7)	e) Dieses Video war eine richtige Herausforderung für mich. (H)	<input type="radio"/> (1)	<input type="radio"/> (2) <input type="radio"/> (3) <input type="radio"/> (4) <input type="radio"/> (5) <input type="radio"/> (6) <input type="radio"/> (7)	<input type="radio"/> (7)	f) Nach dem Anschauen des Videos erscheint mir der Lernstoff sehr interessant. (I)	<input type="radio"/> (1)	<input type="radio"/> (2) <input type="radio"/> (3) <input type="radio"/> (4) <input type="radio"/> (5) <input type="radio"/> (6) <input type="radio"/> (7)	<input type="radio"/> (7)	g) Ich bin sehr gespannt darauf, wie gut ich bei diesem Video aufgepasst habe. (H)	<input type="radio"/> (1)	<input type="radio"/> (2) <input type="radio"/> (3) <input type="radio"/> (4) <input type="radio"/> (5) <input type="radio"/> (6) <input type="radio"/> (7)	<input type="radio"/> (7)	h) Ich fürchte mich ein wenig davor, dass ich mich bei inhaltlichen Fragen zu diesem Video blamieren könnte. (M)	<input type="radio"/> (1)	<input type="radio"/> (2) <input type="radio"/> (3) <input type="radio"/> (4) <input type="radio"/> (5) <input type="radio"/> (6) <input type="radio"/> (7)	<input type="radio"/> (7)	i) Ich war fest entschlossen, mich bei diesem Video voll anzustrengen. (H)	<input type="radio"/> (1)	<input type="radio"/> (2) <input type="radio"/> (3) <input type="radio"/> (4) <input type="radio"/> (5) <input type="radio"/> (6) <input type="radio"/> (7)	<input type="radio"/> (7)	j) Bei einem Video wie diesem brauche ich keine Belohnung, es macht mir auch so Spaß. (I)	<input type="radio"/> (1)	<input type="radio"/> (2) <input type="radio"/> (3) <input type="radio"/> (4) <input type="radio"/> (5) <input type="radio"/> (6) <input type="radio"/> (7)	<input type="radio"/> (7)	k) Es ist mir etwas peinlich, bei inhaltlichen Fragen zu diesem Video zu versagen. (M)	<input type="radio"/> (1)	<input type="radio"/> (2) <input type="radio"/> (3) <input type="radio"/> (4) <input type="radio"/> (5) <input type="radio"/> (6) <input type="radio"/> (7)	<input type="radio"/> (7)	l) Ich glaube mit Hilfe dieses Videos kann jeder lernen. (E)	<input type="radio"/> (1)	<input type="radio"/> (2) <input type="radio"/> (3) <input type="radio"/> (4) <input type="radio"/> (5) <input type="radio"/> (6) <input type="radio"/> (7)	<input type="radio"/> (7)	m) Ich glaube, ich kann mit diesem Video nicht lernen. (E)	<input type="radio"/> (1)	<input type="radio"/> (2) <input type="radio"/> (3) <input type="radio"/> (4) <input type="radio"/> (5) <input type="radio"/> (6) <input type="radio"/> (7)	<input type="radio"/> (7)	n) Wenn ich mit Hilfe dieses Videos etwas lerne, werde ich schon ein wenig stolz auf meine Tüchtigkeit sein. (H)	<input type="radio"/> (1)	<input type="radio"/> (2) <input type="radio"/> (3) <input type="radio"/> (4) <input type="radio"/> (5) <input type="radio"/> (6) <input type="radio"/> (7)	<input type="radio"/> (7)	o) Wenn ich an dieses Video denke, bin ich etwas beunruhigt. (M)	<input type="radio"/> (1)	<input type="radio"/> (2) <input type="radio"/> (3) <input type="radio"/> (4) <input type="radio"/> (5) <input type="radio"/> (6) <input type="radio"/> (7)	<input type="radio"/> (7)	p) Mit dieser Art von Video würde ich mich auch in meiner Freizeit beschäftigen. (I)	<input type="radio"/> (1)	<input type="radio"/> (2) <input type="radio"/> (3) <input type="radio"/> (4) <input type="radio"/> (5) <input type="radio"/> (6) <input type="radio"/> (7)	<input type="radio"/> (7)	q) Die Leistungsanforderungen aus diesem Video lähmen mich. (M)	<input type="radio"/> (1)	<input type="radio"/> (2) <input type="radio"/> (3) <input type="radio"/> (4) <input type="radio"/> (5) <input type="radio"/> (6) <input type="radio"/> (7)	<input type="radio"/> (7)
Video -Nr.	VPN - Nr.																																																																										
	Trifft nicht zu	Trifft voll zu																																																																								
(5) a) Ich mag diese Art des Videos. (I)	<input type="radio"/> (1)	<input type="radio"/> (2) <input type="radio"/> (3) <input type="radio"/> (4) <input type="radio"/> (5) <input type="radio"/> (6) <input type="radio"/> (7)	<input type="radio"/> (7)																																																																								
b) Ich glaube, dass ich mit Hilfe des Videos Fragen zur Vorlesungssitzung gewachsen sein werde. (E)	<input type="radio"/> (1)	<input type="radio"/> (2) <input type="radio"/> (3) <input type="radio"/> (4) <input type="radio"/> (5) <input type="radio"/> (6) <input type="radio"/> (7)	<input type="radio"/> (7)																																																																								
c) Wahrscheinlich werde ich inhaltliche Fragen zur Vorlesungssitzung, die auf dem Video basieren, nicht schaffen. (E)	<input type="radio"/> (1)	<input type="radio"/> (2) <input type="radio"/> (3) <input type="radio"/> (4) <input type="radio"/> (5) <input type="radio"/> (6) <input type="radio"/> (7)	<input type="radio"/> (7)																																																																								
d) Ich fühle mich unter Druck bei diesem Video gut aufpassen zu müssen. (M)	<input type="radio"/> (1)	<input type="radio"/> (2) <input type="radio"/> (3) <input type="radio"/> (4) <input type="radio"/> (5) <input type="radio"/> (6) <input type="radio"/> (7)	<input type="radio"/> (7)																																																																								
e) Dieses Video war eine richtige Herausforderung für mich. (H)	<input type="radio"/> (1)	<input type="radio"/> (2) <input type="radio"/> (3) <input type="radio"/> (4) <input type="radio"/> (5) <input type="radio"/> (6) <input type="radio"/> (7)	<input type="radio"/> (7)																																																																								
f) Nach dem Anschauen des Videos erscheint mir der Lernstoff sehr interessant. (I)	<input type="radio"/> (1)	<input type="radio"/> (2) <input type="radio"/> (3) <input type="radio"/> (4) <input type="radio"/> (5) <input type="radio"/> (6) <input type="radio"/> (7)	<input type="radio"/> (7)																																																																								
g) Ich bin sehr gespannt darauf, wie gut ich bei diesem Video aufgepasst habe. (H)	<input type="radio"/> (1)	<input type="radio"/> (2) <input type="radio"/> (3) <input type="radio"/> (4) <input type="radio"/> (5) <input type="radio"/> (6) <input type="radio"/> (7)	<input type="radio"/> (7)																																																																								
h) Ich fürchte mich ein wenig davor, dass ich mich bei inhaltlichen Fragen zu diesem Video blamieren könnte. (M)	<input type="radio"/> (1)	<input type="radio"/> (2) <input type="radio"/> (3) <input type="radio"/> (4) <input type="radio"/> (5) <input type="radio"/> (6) <input type="radio"/> (7)	<input type="radio"/> (7)																																																																								
i) Ich war fest entschlossen, mich bei diesem Video voll anzustrengen. (H)	<input type="radio"/> (1)	<input type="radio"/> (2) <input type="radio"/> (3) <input type="radio"/> (4) <input type="radio"/> (5) <input type="radio"/> (6) <input type="radio"/> (7)	<input type="radio"/> (7)																																																																								
j) Bei einem Video wie diesem brauche ich keine Belohnung, es macht mir auch so Spaß. (I)	<input type="radio"/> (1)	<input type="radio"/> (2) <input type="radio"/> (3) <input type="radio"/> (4) <input type="radio"/> (5) <input type="radio"/> (6) <input type="radio"/> (7)	<input type="radio"/> (7)																																																																								
k) Es ist mir etwas peinlich, bei inhaltlichen Fragen zu diesem Video zu versagen. (M)	<input type="radio"/> (1)	<input type="radio"/> (2) <input type="radio"/> (3) <input type="radio"/> (4) <input type="radio"/> (5) <input type="radio"/> (6) <input type="radio"/> (7)	<input type="radio"/> (7)																																																																								
l) Ich glaube mit Hilfe dieses Videos kann jeder lernen. (E)	<input type="radio"/> (1)	<input type="radio"/> (2) <input type="radio"/> (3) <input type="radio"/> (4) <input type="radio"/> (5) <input type="radio"/> (6) <input type="radio"/> (7)	<input type="radio"/> (7)																																																																								
m) Ich glaube, ich kann mit diesem Video nicht lernen. (E)	<input type="radio"/> (1)	<input type="radio"/> (2) <input type="radio"/> (3) <input type="radio"/> (4) <input type="radio"/> (5) <input type="radio"/> (6) <input type="radio"/> (7)	<input type="radio"/> (7)																																																																								
n) Wenn ich mit Hilfe dieses Videos etwas lerne, werde ich schon ein wenig stolz auf meine Tüchtigkeit sein. (H)	<input type="radio"/> (1)	<input type="radio"/> (2) <input type="radio"/> (3) <input type="radio"/> (4) <input type="radio"/> (5) <input type="radio"/> (6) <input type="radio"/> (7)	<input type="radio"/> (7)																																																																								
o) Wenn ich an dieses Video denke, bin ich etwas beunruhigt. (M)	<input type="radio"/> (1)	<input type="radio"/> (2) <input type="radio"/> (3) <input type="radio"/> (4) <input type="radio"/> (5) <input type="radio"/> (6) <input type="radio"/> (7)	<input type="radio"/> (7)																																																																								
p) Mit dieser Art von Video würde ich mich auch in meiner Freizeit beschäftigen. (I)	<input type="radio"/> (1)	<input type="radio"/> (2) <input type="radio"/> (3) <input type="radio"/> (4) <input type="radio"/> (5) <input type="radio"/> (6) <input type="radio"/> (7)	<input type="radio"/> (7)																																																																								
q) Die Leistungsanforderungen aus diesem Video lähmen mich. (M)	<input type="radio"/> (1)	<input type="radio"/> (2) <input type="radio"/> (3) <input type="radio"/> (4) <input type="radio"/> (5) <input type="radio"/> (6) <input type="radio"/> (7)	<input type="radio"/> (7)																																																																								

Denken Sie bitte noch einmal an das angeschaute Video zurück. Nachstehend geht es uns **nicht um den Inhalt des Videos**. Vielmehr interessiert uns wie Sie die **Art des Videos** empfunden haben. Kreuzen Sie bitte die **Zahl** zu den einzelnen Aussagen in jeder Zeile an, die am Besten auf Sie zutrifft.

(6)

	Trifft nicht zu		Trifft voll zu
a) Ich konnte dem Video aufmerksam folgen.	<input type="radio"/> (1)	<input type="radio"/> (2)	<input type="radio"/> (3) <input type="radio"/> (4)
b) Ich habe mich von diesem Video leicht ablenken lassen.	<input type="radio"/> (1)	<input type="radio"/> (2)	<input type="radio"/> (3) <input type="radio"/> (4)
c) Ich habe mich während des Videos mit anderen Dingen beschäftigt.	<input type="radio"/> (1)	<input type="radio"/> (2)	<input type="radio"/> (3) <input type="radio"/> (4)
d) Ich fand die Art des Videos ermüdend.	<input type="radio"/> (1)	<input type="radio"/> (2)	<input type="radio"/> (3) <input type="radio"/> (4)
e) Ich schaute mir das Video konzentriert an.	<input type="radio"/> (1)	<input type="radio"/> (2)	<input type="radio"/> (3) <input type="radio"/> (4)
f) Während des Videos habe ich keine Langeweile empfunden.	<input type="radio"/> (1)	<input type="radio"/> (2)	<input type="radio"/> (3) <input type="radio"/> (4)

Stellen Sie sich vor, in der Art des angeschauten Videos werden Ihre Vorlesungen, in denen Sie am Ende des Semesters eine Klausur schreiben müssen, aufgezeichnet und online zum download bereitgestellt. Bitte kreuzen Sie zu jeder Aussage Ihre Meinung an.

(7)

	Trifft nicht zu		Trifft voll zu
a) Ich kann mir nicht vorstellen mit dieser Art von Video für die Klausur zu lernen.	<input type="radio"/> (1)	<input type="radio"/> (2)	<input type="radio"/> (3) <input type="radio"/> (4)
b) Ich würde lieber die Vorlesung besuchen anstatt mit dieser Art von Video zu lernen.	<input type="radio"/> (1)	<input type="radio"/> (2)	<input type="radio"/> (3) <input type="radio"/> (4)
c) Ich würde diese Art des Videos als Zusatzangebot zur Vorlesung nutzen.	<input type="radio"/> (1)	<input type="radio"/> (2)	<input type="radio"/> (3) <input type="radio"/> (4)
d) Eine Vorlesung kann nicht durch diese Art von Video ersetzt werden.	<input type="radio"/> (1)	<input type="radio"/> (2)	<input type="radio"/> (3) <input type="radio"/> (4)
e) Ich würde es begrüßen, wenn es statt der Vorlesung nur noch diese Art von Videos geben würde.	<input type="radio"/> (1)	<input type="radio"/> (2)	<input type="radio"/> (3) <input type="radio"/> (4)
f) Ich würde mir diese Art des Videos nur anschauen, wenn ich zum Vorlesungstermin verhindert wäre.	<input type="radio"/> (1)	<input type="radio"/> (2)	<input type="radio"/> (3) <input type="radio"/> (4)

Nachstehend interessiert uns, warum Sie sich dieses Vorlesungsvideo anschauen. Bitte kreuzen Sie zu jeder Aussage Ihre Meinung an.

(8) Ich schaute mir dieses Video an, weil ...

	Trifft nicht zu		Trifft voll zu
a) ... ich für die Teilnahme entlohnt werde.	<input type="radio"/> (1)	<input type="radio"/> (2)	<input type="radio"/> (3) <input type="radio"/> (4)
b) ... ich durch meine Teilnahme die Forschung in diesem Bereich unterstützen wollte.	<input type="radio"/> (1)	<input type="radio"/> (2)	<input type="radio"/> (3) <input type="radio"/> (4)
c) ... der Vorlesungsinhalt für meine zukünftige Berufslaufbahn bedeutend ist.	<input type="radio"/> (1)	<input type="radio"/> (2)	<input type="radio"/> (3) <input type="radio"/> (4)
d) ... mich die Technik anhand derer der Lernstoff dargeboten wird interessiert.	<input type="radio"/> (1)	<input type="radio"/> (2)	<input type="radio"/> (3) <input type="radio"/> (4)
e) ... ich gerne multimedial lerne.	<input type="radio"/> (1)	<input type="radio"/> (2)	<input type="radio"/> (3) <input type="radio"/> (4)

Nun möchten wir noch ein kurzes Statement von Ihnen allgemein zu diesem Video.

(9)

	1 mal	zwischen 2 - 3 mal	zwischen 4 - 5 mal	öfter als 5 mal
a) Wie oft müssten Sie sich dieses Video ansehen, um den vollständigen Lernstoff der Vorlesungssitzung zu erfassen?	<input type="radio"/> (1)	<input type="radio"/> (2)	<input type="radio"/> (3)	<input type="radio"/> (4)
b) Wie viel haben Sie Ihrer Meinung nach durch diese Art der Videoführung in der Vorlesung gelernt.	<input type="radio"/> (1)	<input type="radio"/> (2)	<input type="radio"/> (3)	<input type="radio"/> (4)

(5) Möchten Sie uns noch etwas zu dieser Art des Videos mitteilen?

7.3.3. German Post-Test Form

FAKULTÄT FÜR SOZIALWISSENSCHAFTEN
Lehrstuhl Erziehungswissenschaft I
Prof. Dr. Peter Drewek

INSTITUT FÜR INFORMATIK
Lehrstuhl für Praktische Informatik IV
Prof. Dr. Wolfgang Effelsberg

UNIVERSITÄT
MANNHEIM

Video-Nr.	VPN-Nr.
-----------	---------

Nachfragen zum Video

Die Beantwortung der nachstehenden Fragen zur Ermittlung Ihrer Kenntnisse im Themengebiet „Audioaufnahme und Audioschnitte für Videoproduktionen“ erfolgen **anonym** und dienen uns lediglich zur Erfassung Ihres Wissens in diesem Bereich.

- Was ist der Unterschied zwischen der Samplingtiefe und der Samplerate?
- In welcher Einheit wird die Samplingtiefe angegeben?
- Nennen Sie die typischen Eigenschaften von qualitativ hochwertigen DV-Video.
- Nennen Sie zwei verschiedene Mikrofon-Charakteristiken.
- Was benötigen Elektret-Kondensatormikrofone?
- Was ist der Unterschied zwischen einem Popp-Schutz und einem Windschutz?
- Wie viele interviewte Personen sollen maximal mit einem Mikrofon in einem Interview aufgenommen werden?

- Was ist der Unterschied zwischen einer Mikrofonangel und einem Mikrofonstativ?
- Was ist eine Atmo-Spur?
- Wozu wird eine Atmo-Spur benötigt?
- Wie heisst der besprochene Effekt, der eintreten kann, wenn ein Signal mit mehreren Mikrofonen aufgenommen wird?
- Wie laut soll der lauteste Ton aufgenommen werden? (in dB)
- Wie laut soll der leiseste, gut hörbare, Ton aufgenommen werden? (in dB)
- Nennen Sie ein Beispiel, wann eine Ton-Überblendung bei einem Szenenwechsel sinnvoller ist, als ein harter Schnitt?

Vielen Dank für Ihre Teilnahme!

8. Bibliography

- [Amft *et al.*, 2004] Amft, O., Lauffer, M., Ossevoort, S., Macaluso, F., Lukowicz, P., Troster, G.: *Design of the QBIC wearable computing platform*, Proceedings on Application-specific Systems, Architectures and Processors, 2004, IEEE Computer Society, Galveston, Texas, 398-410.
- [Baecker, 2003] Baecker, R.: *A principled design for scalable internet visual communications with rich media, interactivity, and structured archives*, Proceedings of the 2003 conference of the Centre for Advanced Studies on Collaborative research 2003, Toronto, Ontario, Canada, 16-29.
- [Bär *et al.*, 2005] Bär, H., Mühlhäuser, M., Tews, E. & Rößling, G.: *Interaction During Lectures Using Mobile Phones*. Proceedings of World Conference on Educational Multimedia, Hypermedia and Telecommunications (Ed-Media) 2005, Chesapeake, VA, USA, 3767-3772.
- [Benz, 2007] Benz, M.: *Erstellung eines Kameramann-Moduls für Automatisierte Vorlesungsaufzeichnungen*, Diplomarbeit, Fakultät für Mathematik und Informatik, Praktische Informatik IV, Universität Mannheim, Mannheim, 2007.
- [Bianchi, 1998] Bianchi, M.: *AutoAuditorium: a fully automatic, multicamera system to televise auditorium presentations*, Proceedings of the Joint DARPA/NIST workshop on smart spaces technology 1998, Gaithersburg, MD.
- [Bianchi, 2004] Bianchi, M.: *Automatic video production of lectures using an intelligent and aware environment*, Proceedings of the 3rd international conference on Mobile and ubiquitous multimedia 2004, College Park, Maryland, USA, 117-123.
- [Brauer, 1984] Brauer, W.: *Automatentheorie*, Vieweg + Teubner Verlag, Stuttgart, 1984, ISBN 978-3519022510.
- [Bortz & Döring, 2006] Bortz, J, Döring, N.: *Forschungsmethoden und Evaluation für Human- und Sozialwissenschaftler*, 4. Auflage, Springer Verlag, Berlin, 2006, ISBN 978-3540333050.

- [Brotherton, 2001] Brotherton, J.: *Enriching Everyday Activities through the Automated Capture and Access of Live Experiences; eClass: Building, Observing and Understanding the Impact of Capture and Access in an Educational Domain*. Ph.D. Thesis, Georgia Institute of Technology, 2001.
- [Büren von, 2002] Büren von, Th.: *Extended Location and Environment Condition Sensor*, Wearable Computing Laboratory IfE, ETH Zürich, Zürich, 2002.
- [Camtasia, 2009] Camtasia/Techsmith: *Camtasia Homepage*, <http://www.techsmith.de/camtasia.asp?>, last visited: 04. April 2010.
- [Choi *et al.*, 2004] Choi, C., Bär, H., Röbling, G., Mühlhäuser, M.: *Elektronische Interaktion in großen Lehrveranstaltungen*, Proceedings of Informatik 2004, GI-Edition: Proceedings Vol.50, Bd.1, Bonn, Germany, 419-423.
- [Christianson *et al.*, 1996] Christianson, D.B., Anderson, S.E., He, L., Salesin, D.H., Weld, D.S., Cohen, M.F.: *Declarative Camera Control for Automatic Cinematography*, Proceedings of the 13th National Conference on Artificial Intelligence, AAAI '96, Portland, OR, USA, 148-155.
- [Courty *et al.*, 2003] Courty, N., Lamarche, F., Donikian, St., Marchand, É., *A Cinematography System for Virtual Story Telling*, Proceedings of ICVS 2003, Springer Verlag, LNCS 2897, 30-34.
- [Cromie, 2006] Cromie, J.: *QuickTime for .NET and COM Developers*, Elsevier Morgan Kaufmann Publishers, San Francisco, 2006.
- [Cutler *et al.*, 2002] Cutler, R., Rui, Y., Gupta, A., Cadiz, J.J., Tashev, I., He, L., Colburn, A., Zhang, Z., Liu, Z., Silverberg, St., *Distributed Meetings: A Meeting Capture and Broadcasting System*, Proceedings of ACM Multimedia 2002, Juan-les-Pins, France, 503-512.
- [Cruz & Hill, 1994] Cruz, G., Hill, R.: *Capturing and playing multimedia events with STREAMS*, Proceedings of the second ACM international conference on Multimedia 1994, San Francisco, California, USA, 193-200.
- [Dal Lago *et al.*, 2002] Dal Lago, S., De Petris, G., Pigazzini, P., Sarti, A., Tubaro, S.: *Real-time Content Creation For Multimedia Didactics*, 4th International Conference on New Educational Environments ICNEE 2002, Lugano, Switzerland.
- [Datta & Ottmann, 2001] Datta, A., Ottmann, Th.: *Towards a Virtual University*, Journal of Universal Computer Science, Vol.7, No.10, 2001, 870-885.

- [**Dickreiter et al., 2008a**] Dickreiter, M., Hoeg, W., Dittel, V., Wöhr, M.: *Handbuch der Tonstudiotechnik Band 1*, 7. völlig neu bearbeitete und erweiterte Auflage, K.G. Saur Verlag, München 2008.
- [**Dickreiter et al., 2008b**] Dickreiter, M., Hoeg, W., Dittel, V., Wöhr, M.: *Handbuch der Tonstudiotechnik Band 2*, 7. völlig neu bearbeitete und erweiterte Auflage, K.G. Saur Verlag, München 2008.
- [**DirectShowTutorial, 2009**] Microsoft: *Writing DirectShow Filters Homepage*, <http://msdn.microsoft.com/en-us/library/dd391013%28VS.85%29.aspx>, last visited: 04. April 2010.
- [**Effelsberg, 1998**] Effelsberg, W.: *Vorlesung Rechnernetze Wintersemester 98/99*, <http://pi4.informatik.uni-mannheim.de/pi4.data/content/courses/1998-ws/rn9899/Vorlesung/rn10-1.pdf>, last visited: 07. April 2010.
- [**FFmpeg, 2009**] FFmpeg™ of Bellard, F.: *FFmpeg project homepage*, <http://ffmpeg.org>, last visited: 04. April 2010.
- [**Friedland, 2006**] Friedland, G.: *Adaptive Audio and Video Processing for Electronic Chalkboard Lectures*, Dissertation, Fachbereich Mathematik und Informatik, Freie Universität Berlin, Berlin 2006.
- [**Friedland, Jantz & Knipping, 2004**] Friedland, G., Jantz, K., Knipping, L.: *Towards automatized studioless audio recording: a smart lecture recorder*. TechReport, Department of Mathematics and Computer Science, FU Berlin, 2004, <http://www.inf.fu-berlin.de/inst/ag-ki/ger/b-04-14.pdf>, last visited 04. April 2010.
- [**Friedland et al., 2005**] Friedland, G., Jantz, K., Knipping, L., Rojas, R.: *The Virtual Technician: An Automatic Software Enhancer for Audio Recording in Lecture Halls*, In: Knowledge-Based Intelligent Information and Engineering Systems, LNCS 3681, Springer Verlag Berlin Heidelberg, 2005, 744-750.
- [**Friedland & Pauls, 2005**] Friedland, G., Pauls, K.: *Architecting multimedia environments for teaching*, IEEE Computer Journal, Vol. 38, No.6, 2005, 57-64.
- [**Funkkolleg, 2009**] Funkkolleg, Hessischer Rundfunk, hr2, Frankfurt am Main, <http://www.funkkolleg.de>, last visited 04. April 2010.

- [Gleicher & Masanz, 2000] Gleicher, M., Masanz, J.: *Towards Virtual Videography*, Proceedings of ACM Multimedia 2000, Marina del Rey, CA, USA, 375-378.
- [Gleicher, Heck & Wallick, 2002] Gleicher, M., Heck, R., Wallick, M.: *A Framework for Virtual Videography*, ACM international Conference Proceedings Series; Vol.24, Proceedings of 2nd international symposium on Smart graphics, Hawthorne, 2002, NY, USA, 9-16.
- [Häussge *et al.*, 2008] Häussge, G., Hartle, M., Neziri, A., Rößling, G.: *Plug'n'Present: Eine referentenorientierte Infrastruktur zur Präsentation und Aufzeichnung von Vorlesungen.*, Proceedings of DeLFI 2008: Die 6. e-Learning Fachtagung Informatik der Gesellschaft für Informatik e.V., GI LNI, Vol.132, 2008, 161-172.
- [Hampapur *et al.*, 2005] Hampapur, A., Brown, L., Connell, J., Ekin, A., Haas, N., Lu, M., Merkl, H., Pankanti, S., *Smart video surveillance: exploring the concept of multiscale spatiotemporal tracking*, IEEE Signal Processing Magazine 03/2005, Vol. 22, No. 2, 38-51.
- [Hartle *et al.*, 2005] Hartle, M., Bär, H., Trompler, Chr., Rößling, G., *Perspectives for Lecture Videos*, Proceedings of Euro-Par 2005 Parallel Processing, Springer Verlag, LNCS 3648, 901-908.
- [He, Cohen & Salesin, 1996] He, L., Cohen, M.F., Salesin, D.H.: *The virtual cinematographer: A paradigm for automatic real-time camera control and directing*, Proceedings of ACM SIGGRAPH: 23. International Conference on Computer Graphics and Interactive Training 1996, 217-224.
- [He, Grudin & Gupta, 2000] He, L., Grudin, J., Gupta, A.: *Designing presentations for on-demand viewing*, Proceedings of the 2000 ACM conference on Computer supported cooperative work, Philadelphia, Pennsylvania, USA, 127-134.
- [He & Zhang, 2007] He, L., Zhang, Z.: *Real-Time White-board Capture and Processing Using a Video Camera for Remote Collaboration*, IEEE Transactions on Multimedia, Vol.9, No.1, 2007, 198-206.
- [Heck, Wallick & Gleicher, 2007] Heck, R., Wallick, M., Gleicher, M.: *Virtual Videography*, ACM Transactions on Multimedia Computing, Communications, and Applications (TOMCCAP), Vol.3, No.1, 2007, Article No.4.

- [Herweh, 2009] Herweh, P.: *Ein intuitives Meldesystem für die Vorlesungsaufzeichnung*, Diplomarbeit, Fakultät für Mathematik und Informatik, Universität Mannheim, Mannheim, 2009.
- [Huang, Cui & Samarasekera, 1998] Huang, Q., Cui, Y., Samarasekera, S.: *Content based active video data acquisition via automated cameramen*, Proceedings of IEEE International Conference on Image Processing ICIP 1998, 808-812.
- [Hürst et al., 2001] Hürst, W., Maass, G., Müller, R., Ottmann, Th.: *The "Authoring on the Fly" system for automatic presentation recording*, Proceedings of Conference on Human Factors in Computing Systems CHI 2001, Seattle, Washington, USA, 5-6.
- [Hürst, Müller & Ottmann, 2004] Hürst, W., Müller, R., Ottmann, Th.: *The AOF Method for Production, Use, and Management of Instructional Media*. Proceedings of International Conference on Computers in Education ICCE 2004, Melbourne, Australia, <http://citeseer.ist.psu.edu/729098.html>, last visited: 04. April 2010.
- [King, Kopf & Effelsberg, 2005] King, Th., Kopf, St., Effelsberg, W.: *A Location System based on Sensor Fusion: Research Areas and Software Architecture*, Proceedings of 2. GI/ITG KuVS Fachgespräch "Ortsbezogene Anwendungen und Dienste", 28-32.
- [King et al., 2007] King, Th., Haenselmann, Th., Kopf, St., Effelsberg, W.: *Overhearing the Wireless Interface for 802.11-based Positioning Systems*, Proceedings of 5th Annual IEEE International Conference on Pervasive Computing and Communications (PerCom 2007), New York, USA, 145-150.
- [Kopf et al., 2003] Kopf, St., Haenselmann, Th., Farin, D., Effelsberg, W.: *Automatic Generation of Summaries for the Web*, Proceedings of the Society of Photo-Optical Instrumentation Engineers (SPIE), Vol. 5307, 2003, 417-428.
- [Kopf, Haenselmann & Effelsberg, 2004] Kopf, St., Haenselmann, Th., Effelsberg, W.: *Shape-based Posture and Gesture Recognition in Videos*, Proceedings of the Society of Photo-Optical Instrumentation Engineers (SPIE), Vol. 5682, 2004, 114-124.

- [Kopf & Effelsberg, 2007] Kopf, St., Effelsberg, W.: *New Teaching and Learning Technologies for Interactive Lectures*, Journal of Advanced Technology for Learning, ActaPress, Calgary, Canada, Vol. 4, No. 2, 2007, 60-67.
- [Konerow, 2007] Konerow, J.: *Managed DirectX und C#- Einstieg und professioneller Einsatz*, Software & Support Verlag, entwickler.press, Frankfurt am Main, 2007.
- [Kuhmünch, 2001] Kuhmünch, Chr.: *Videoskalierung und Integration interaktiver Elemente in Teleteaching Szenarien*, Dissertation, Akademische Verlagsgesellschaft Aka GmbH, Berlin, 2001.
- [Lampi, Kopf & Effelsberg, 2006] Lampi, F., Kopf, St., Effelsberg, W.: *Mediale Aufbereitung von Lehrveranstaltungen und ihre automatische Veröffentlichung - Ein Erfahrungsbericht*, Proceedings of DeLFI 2006, 4. e-Learning Fachtagung Informatik der Gesellschaft für Informatik, Darmstadt, Germany, 2006, 27-38.
- [Lampi, Kopf & Effelsberg, 2008] Lampi, F., Kopf, St., Effelsberg, W.: *Automatic Lecture Recording*, Proceedings of ACM Multimedia 2008, Vancouver, BC, Canada, 1103-1104.
- [Lampi et al., 2009] Lampi, F., Lemelson, H., Kopf, St., Effelsberg, W.: *A Question managing suite for automatic lecture recording*, Journal of Interactive Technology and Smart Education, Emerald Group Publishing Limited, Vol. 6, No. 2, 2009, 108-118.
- [Lauer & Ottmann, 2002] Lauer, T., Ottmann, Th.: *Means and Methods in Automatic Courseware Production: Experience and Technical Challenges*, Proceedings of E-Learn 2002, Montreal, Canada, 2002, 553-560.
- [Lemelson, King & Effelsberg, 2008] Lemelson, H., King, Th., Effelsberg, W.: *Pre-processing of Fingerprints to Improve the Positioning Accuracy of 802.11-based Positioning Systems*, Proc. of the First ACM International Workshop on Mobile Entity Localization and Tracking in GPS-less Environments (MELT 2008), San Francisco, United States of America, 2008, 73-78.
- [Leue, 2000] Leue, St.: *Vorlesung Entwurf von Telekommunikationssystemen, Erweiterte Endliche Zustandsmaschinen (EFSMs)*, Albert-Ludwigs-Universität Freiburg, Institut fuer Informatik, Sommersemester 2000,

<http://tele.informatik.uni-freiburg.de/lehre/ss00/etks.teil2/ppframe.htm>,
last visited: 04. April 2010.

- [Liu & Kender, 2004] Liu, T.; Kender, J.R.: *Lecture videos for e-learning: current research and challenges*, Proceedings of IEEE Sixth International Symposium on Multimedia Software Engineering (ISMSE) 2004, 574-578.
- [Liu et al., 2001] Liu, Q., Rui, Y., Gupta, A., Cadiz, J.J.: *Automating Camera Management for Lecture Room Environments*, Proceedings of ACM CHI 2001, Seattle, USA, Vol. 3, 442-449.
- [Liu et al., 2002a] Liu, Q., Kimber, D., Foote, J., Wilcox, L., Boreczky, J.: *FLY-SPEC: a multi-user video camera system with hybrid human and automatic control*, Proceedings of ACM Multimedia 2002, Juan-les-Pins, France, 484-492.
- [Liu et al., 2002b] Liu, Q., Kimber, D., Wilcox, L., Cooper, M., Foote, J., Boreczky, J.: *Managing a Camera System to Serve Different Video Requests*, Proceedings of IEEE International Conference on Multimedia and Expo 2002 (ICME '02), Lausanne, Switzerland, Vol. 2, 13-16.
- [live555, 2009] Live Networks Inc.: *live555 Streaming Media Homepage*, <http://www.live555.com/liveMedia/>, last visited: 04. April 2010.
- [Lukowicz et al., 2002] P. Lukowicz, H. Junker, M. Stäger, T. von Büren, G. Tröster, *WearNET: A Distributed Multi-sensor System for Context Aware Wearables*, Proceedings of the 4th International Conference on Ubiquitous Computing, September 2002, Springer Verlag, 361-370.
- [Ma et al., 2003] Ma, M., Schillings, V., Chen, T., Meinel, Chr.: *T-Cube: A Multimedia Authoring System for eLearning*, Proceedings of AACE E-Learn 2003, Phoenix, AZ, USA, 2289-2296.
- [Machnicki & Rowe, 2002] Machnicki, E., Rowe, L.: *Virtual Director: Automating a Webcast*, Proceedings of SPIE Multimedia Computing and Networking 2002, San Jose, CA, USA, 208-225.
- [Matsuo, Amano & Uehara, 2002] Matsuo, Y., Amano, M., Uehara, K.: *Mining video editing rules in video streams*, Proceedings of the tenth ACM international conference on Multimedia 2002, Juan-les-Pins, France, 255-258.

- [Mertens & Rolf, 2003] Mertens, R., Rolf, R.: *Automation Techniques for Broadcasting and Recording Lectures and Seminars*, eProceedings of 3rd International Technical Workshop and Conference SINN '03, Universität Oldenburg, Institute for Science Networking Oldenburg, http://physnet.uni-oldenburg.de/projects/SINN/sinn03/proceedings/mertens_rolf.html, last visited 04. April 2010.
- [Millerson, 1990] Millerson, G.: *Die Videokamera. Technik - Einsatzgebiete - Bildgestaltung*. 1. Auflage. ARES Enterprises, Köln, 1990.
- [MoCA, 2006] MoCA-project: The Movie Content Analysis Project, Project homepage, University of Mannheim, Praktische Informatik IV, <http://pi4.informatik.uni-mannheim.de/pi4.data/content/projects/moca/>, last visited 04. April 2010.
- [Monaco, 2000a] Monaco, J.: *How to Read a Film –Movies, Media, Multimedia*, 3rd edition, Oxford University Press, New York, 2000.
- [Monaco, 2000b] Monaco, J.: *How to Read a Film* on How to Read a Film, Multimedia Edition DVD-ROM Harbor Electronic Publishing New York and Sag Harbor, 2000.
- [Mühlhäuser, 2005] Mühlhäuser, M.: *Digitale Hörsäle: wo Präsenz- und Cyber-Universität sich treffen.*, In: „Studieren im Cyberspace? : die Ausweitung des Campus in den virtuellen Raum“, Lit-Verlag, 2005, ISBN: 3-8258-8420-1, 31-44.
- [Müller & Ottmann, 2000] Müller, R., Ottmann, Th.: *The “Authoring on the Fly” system for automated recording and replay of (tele)presentations*, Multimedia Systems Journal, Springer Verlag Berlin - Heidelberg, Vol.8, No.3, 2000, 158-176.
- [Müller, Ottmann & Zhang, 2002] Müller, R., Ottmann, Th., Zhang, H.: *Presentation Recording as a means to go virtual for Campus-based Universities*, Proceedings of IRMA 2002, Seattle, WA, USA.
- [Mukhopadhyay & Smith, 1999] Mukhopadhyay, S., Smith, B.: *Passive capture and Structuring of Lectures*, Proceedings of ACM Multimedia 1999, Orlando, FL, USA, Vol.: 1, 477-487.

- [Onishi & Fukunaga, 2004] Onishi, M., Fukunaga, K.: *Shooting the lecture scene using computer-controlled cameras based on situation understanding and evaluation of video images*, Proceedings of the 17th International Conference on Pattern Recognition (ICPR) 2004, Vol.1, 781-784.
- [Open2.net, 2009] *Open2.net: Online Learning Portal*, The Open University and the BBC, Milton Keynes, UK, <http://www.open2.net/>, last visited: 04. April 2010.
- [Pospeschill, 2006] Pospeschill, M.: *Statistische Methoden: Strukturen, Grundlagen, Anwendungen in Psychologie und Sozialwissenschaften*, Spektrum Akademischer Verlag, 2006, ISBN 978-3827415509.
- [Rensing et al., 2008] Rensing, Ch., Zimmermann, B., Meyer, M., Lehmann, L., Steinmetz, R.: *Wiederverwendung von multimedialen Lernressourcen im Re-Purposing und Authoring by Aggregation*, In Prozessorientiertes Authoring Management: Methoden, Werkzeuge und Anwendungsbeispiele für die Erstellung von Lerninhalten, Logos Verlag, Berlin, 2008, 19-40.
- [Rheinberg, Vollmeyer & Burns, 2001] Rheinberg, F., Vollmeyer, R., Burns, B.D.: *FAM: Ein Fragebogen zur Erfassung aktueller Motivation in Lern- und Leistungssituationen.*, Diagnostica, Vol. 47, 2001, 57-66.
- [Rößling & Ackermann, 2007] Rößling, G., Ackermann, T.: *A Framework for Generating AV Content on-the-fly*, Electronic Notes in Theoretical Computer Science (ENTCS) Journal, Vol.178, Elsevier Science Publishers B. V. Amsterdam, The Netherlands, 23-31.
- [Rößling et al., 2006] Rößling, G., Bär, H., Turban, G., Hartle, M., Trompler, Chr., Mühlhäuser, M.: *Digitale Hörsäle als Wegbereiter bei der Evolution einer klassischen Lehrveranstaltung hin zum E-Learning*, Proceedings of DeLFI 2006, 4. E-Learning Fachtagung Informatik, Darmstadt, Germany, GI-Edition - Lecture Notes in Informatics (LNI), P-87, 387-388.
- [Rohde & Schwarz, 2006] Winter, A., for Rohde & Schwarz: *dB or not dB? Was Sie schon immer zum Rechnen mit dB wissen wollten...*, Application Note 1MA98, Rohde & Schwarz, http://www2.rohde-schwarz.com/file_6407/1MA98_4D.pdf, last visited: 04. April 2010.
- [Rowe et al., 2003] Rowe, L., Harley, D., Pletcher, P., Lawrence, S.: *BIBS: A Lecture Webcasting System*, TechReport of Center for Studies in Higher Educa-

tion, University of California at Berkeley,
<http://ideas.repec.org/p/cdl/cshedu/1005.html>,
 last visited: 04. April 2010.

[Rowe & Casalaina, 2006] Rowe, L., Casalaina, V.: *Capturing Conference Presentations*, IEEE Multimedia Journal, Vol.13, No.4, 2006, 76-84.

[Rui et al., 2001] Rui, Y., He, L. Gupta, A., Liu, Q.: *Building an intelligent camera management system*. Proceedings of ACM Multimedia 2001, Ottawa, Canada, 2-11.

[Rui et al., 2004] Rui, Y., Gupta, A., Grudin, J., He, L.: *Automating lecture capture and broadcast: technology and videography*. Multimedia Systems Journal Vol.10, No.1, 2004, Springer Verlag, 3-15.

[Rui & Florencio, 2004] Rui, Y.; Florencio, D.: *Time delay estimation in the presence of correlated noise and reverberation*, Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP) 2004, Vol.2, ii-133-6.

[Rui, Gupta & Cadiz, 2001] Rui, Y., Gupta, A., Cadiz, J.J.: *Viewing meetings captured by an omni-directional camera*, Proceedings of ACM CHI 2001, Seattle, WA, USA, 450-457.

[Rui, Gupta & Grudin, 2003] Rui, Y., Gupta, A., Grudin, J.: *Videography for Telepresentations*, Proceedings of ACM CHI 2003, Fort Lauderdale, FL, USA, 457-464.

[Scheele et al., 2003] Scheele, N., Mauve, M., Effelsberg, W., Wessels, A., Horz, H., Fries, S.: *The Interactive Lecture - A new Teaching Paradigm Based on Ubiquitous Computing*, Proceedings of Computer Supported Collaborative Learning CSCL 2003, Bergen, Norway, 135-137.

[Scheele et al., 2004] Scheele, N., Seitz, C., Effelsberg, W., Wessels, A.: *Mobile devices in Interactive Lectures*, Proceedings of the World Conference on Educational Multimedia, Hypermedia & Telecommunications (ED-MEDIA '04), Lugano, Switzerland, 154-161.

[Scheele et al., 2005] Scheele, N., Wessels, A., Effelsberg, W., Hofer, M., Fries, St.: *Experiences with Interactive Lectures - Considerations from the Perspective of Educational Psychology and Computer Science*, Proceedings of the

- International Conference on Computer Supported Collaborative Learning
2005, Taipeh, Taiwan, 547-556.
- [Schmidt, 2005] Schmidt, U.: *Professionelle Videotechnik, 4. aktualisierte und erweiterte Auflage*, Springer-Verlag, Berlin – Heidelberg, 2005.
- [Schult & Buchholz, 2002] Schult, G., Buchholz, A.: *Fernseh-Journalismus, Ein Handbuch für Ausbildung und Praxis*, 6. aktualisierte Auflage, List Verlag, München, 2002.
- [Scott & Mason, 2001] Scott, P., Mason, R.: *Graduating live and on-line: The multimedia webcast of the open university's worldwide virtual degree ceremony*, Proceedings of AACE WebNet 2001, Orlando, Florida, USA, 1095-1100.
- [Shi et al., 2003] Shi, Y., Xie, W., Xu, G., Shi, R., Chen, E., Mao, Y., Liu, F.: *The Smart Classroom: Merging Technologies for Seamless Tele-Education*, IEEE Pervasive Computing Journal, 2003, Vol.2, No.2, 47-55.
- [TAO-Framework, 2009] A, St., Hudson, D., Loach, R., Ridge, R., Triplett, T.L.: *The TAO Framework Project Homepage*, The TAO Framework, <http://sourceforge.net/projects/taoframework/>, last visited: 04. April 2010
- [Tashev & Malvar, 2005] Tashev, I., Malvar, H.S.: *A new beamformer design algorithm for microphone arrays*, Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP) 2005, Vol.3, 101-104.
- [Telekolleg, 2009] Telekolleg, Bayrischer Rundfunk, München, <http://www.br-online.de/wissen-bildung/telekolleg/>, last visited 04. April 2010.
- [Thompson, 1993] Thompson, R.: *Grammar of the edit*, Elsevier, Focal Press, Oxford, 1993.
- [Thompson, 1998] Thompson, R.: *Grammar of the shot*, Elsevier, Focal Press, Oxford: 1999.
- [Truong, Abowd & Brotherton, 2001] Truong, K.N., Abowd, G.D., Brotherton, J.A.: *Who, What, When, Where, How: Design Issues of Capture & Access Applications*, Proceedings of Ubicomp 2001: Ubiquitous Computing, Springer Verlag, LNCS 2201, 209-224.
- [Wallick, Rui & He, 2004] Wallick, M., Rui, Y., He, L.: *A Portable Solution for Automatic Lecture Room Camera Management*. Proceedings of IEEE In-

ternational Conference on Multimedia and Expo ICME '04, Vol. 2, 987-990.

[Wang, Ngo & Pong, 2003] Wang, F., Ngo, C.W., Pong, T.C.: *Synchronization of lecture videos and electronic slides by video text analysis*, Proceedings of the eleventh ACM international conference on Multimedia 2003, Berkeley, CA, USA, 315-318.

[WindowsSDK, 2009] Microsoft: *Windows Software Development Kit Homepage*, <http://msdn.microsoft.com/en-us/windows/bb980924.aspx>, last visited: 04. April 2010.

[Yokoi & Fujiyoshi, 2005] Yokoi, T.; Fujiyoshi, H.: *Virtual camerawork for generating lecture video from high resolution images*, Proceedings of IEEE International Conference on Multimedia and Expo (ICME) 2005.

[Zhang et al., 2005] Zhang, C., Rui, Y., He, L., Wallick, M.: *Hybrid Speaker Tracking in An Automated Lecture Room*, Proceedings of IEEE International Conference on Multimedia and Expo ICME '05, 4 pp.

[Zhang et al., 2008] Zhang, C., Rui, Y., Crawford, J., He, L.: *An automated end-to-end lecture capture and broadcasting system*, ACM Transactions on Multimedia Computing, Communications, and Applications (TOMCCAP) 2008, Vol.4, No.1, Article 6.

[Ziewer, 2007] Ziewer, P.: *Flexible and Automated Production of Full-Fledged Electronic Lectures*. Dissertation, TU München, Fakultät für Informatik, 2007.

Statement of Originality / Erklärung

Hereby I declare that I wrote this dissertation titled “Automatic Lecture Recording” myself, with the help of no more than the mentioned literature and auxiliary means. This dissertation was not published or presented to another examination office in the same or similar shape.

Hiermit erkläre ich, dass ich die vorliegende Inauguraldissertation mit dem Titel "Automatic Lecture Recording" selbständig und ausschließlich unter Verwendung der angegebenen Literatur und Hilfsmitteln verfasst habe. Die Arbeit wurde in gleicher oder ähnlicher Form keiner anderen Prüfungsbehörde zum Erlangen eines akademischen Grades vorgelegt.

Karlsruhe, den 10.04.2010

Fleming Lampi